

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلوكية و

اللاسلكية

قسم: الإعلام الآلي



Mémoire de Master en informatique

Spécialité : Modélisation informatique des connaissances et raisonnement

(MICR)

Thème

Système de recommandation de lieux utilisant une
approche hybride multicritère



▪ **Présenté par :**

✓ BOUAB Menaouer

▪ **Dirigé par :**

✓ Dr .LATRECHE Abdelkrim

Année universitaire  2024-2025

Remerciements

Nous remercions le bon Allah tout-puissant, qui nous a donné la force, la volonté et le courage pour terminer ce modeste travail. Nous adressons notre profond remerciement à Dr LATRECHE Abdelkrim pour son encadrement, son écoute, ses élucidations, ses conseils, ses directives et encouragements qu'il nous a afflué. Nous remercions Mrs les jurés pour l'intérêt qu'ils ont porté à ce travail en acceptant d'être examinateurs. Ainsi, nous adressons nos remerciements les plus chaleureux à toutes les personnes qui ont aidé de près ou de loin par fruit de leur connaissance pendant toute la durée de notre parcours éducatif. A tous les enseignants dans le département d'informatique Tout simplement à tous ceux et celles qui méritent nos remerciements.

Dédicace

Je voudrais dédier ce travail à ce qui m'est le plus cher après Dieu et le Prophète, paix et salut sur Lui :

À mon père HADJ, qui m'a encouragé à m'instruire et m'a élevé.

À ma chère et la femme la plus proche de mon cœur, ma mère.

Merci à vous deux pour votre amour illimité et votre éducation. J'espère être la meilleure image de vous.

À mes frères LAHCEN, TAHAR et AHMED, et à mon petit frère et ami BOUSMAHA.

À ma sœur aînée, au fils de mon Frère, HADJ, et à la fille de mon frère, Assil.

À tous mes amis : Mohamed, Khalil, Moustafa et Alaa.....

Merci à vous tous.

Et au plus grand soutien que j'ai eu pour ce mémoire : merci à vous, mon cher professeur Dr LATRECHE Abdelkrim

BOUAB MENAOUER

Table de matières

Introduction Générale.....	1
Chapitre 1 : Introduction aux Systèmes de Recommandation	4
1.1 Introduction.....	4
1.2. Historique	4
1.3 Définition des systèmes de recommandation.....	5
1.4 Objectifs des systèmes de recommandation	7
1.5 Classification des systèmes de recommandation.....	8
1.5.1 Classification classique	9
1.5.2 La classification de Su et al.,2009.....	9
1.5.3 La classification de Rao et Talwar, 2008	10
1.6 Etude des algorithmes de recommandations	10
1.6.1 Recommandation basée sur le filtrage collaboratif.....	11
1.6.1.1 Processus du filtrage collaboratif	12
1.6.1.1.1 Evaluation des recommandations	12
1.6.1.1.2 Formation des communautés.....	12
1.6.1.1.3 Production des recommandations	12
1.6.1.1.4 Profils et communautés	12
1.6.1.1.5 Exemple.....	13
1.6.1.2 Approche basée sur la mémoire	14
1.6.1.3 Filtrage collaboratif basé sur un modèle	15
1.6.2. Recommandation basée sur le contenu	17
1.6.2.1- Recommandation basée sur les mots-clés.....	18
1.6.2.2- Recommandation basée sur la sémantique.....	18
1.6.3 Filtrage hybride.....	19
1.7 Autres classifications.....	19
1.7.1 Systèmes de recommandation basés sur la connaissance (KBRS).....	19
1.7.2 Systèmes de recommandation basés sur les ontologies (OBRS)	20
1.7.3 Systèmes de recommandation intelligents (IRS)	20
1.7.4 Systèmes de recommandation basés sur la démographie (DBRS)	20
1.7.5 Systèmes de recommandation sensibles au contexte (CARS)	20
1.8 La recommandation et l'analyse des sentiments.....	21

1.8.1	Approches pour mesurer le sentiment	21
1.8.2	L'approche lexicale	21
1.8.3	L'approche d'apprentissage automatique	22
1.8.4	L'approche hybride	22
1.9	Méthode d'analyse des sentiments VADER	23
1.10	Prédiction de lieux et systèmes de recommandation	23
1.11	Conclusion	24
	Chapitre 02 : Conception	26
2.1	Introduction.....	26
2.2	Architecture générale du système.....	26
2.2.1	Prétraitement.....	27
2.2.1.1	Collection des données	27
2.2.1.2	Analyse des sentiments.....	28
2.2.1.3	Création de la Base de données	29
2.2.2	Fonctionnement du système	30
2.2.3	Algorithme de recommandation	31
2.3	Conception de notre système.....	33
2.3.1	Diagrammes de cas d'utilisation	33
2.3.2	Diagrammes de classe.....	33
2.3.3	Diagrammes de séquence.....	34
2.4	Conclusion	36
	Chapitre 03 : Implémentation et expérimentation	38
3.1	Introduction.....	38
3.2	Environnements de développement et langages utilisés.....	38
3.3	Les Interfaces Graphiques du Système.....	42
3.3.1	Écran de Connexion (Login Screen).....	42
3.3.2	Écran d'Inscription (Register Screen).....	43
3.3.3	Écran du Profil Utilisateur (User Profile Screen)	44
3.3.4	Écran des Catégories (Category Screen).....	45
3.3.5	Écran de Liste des Lieux (Places List Screen)	46
3.3.6	Écran du Journal des Commentaires Utilisateurs (User Comments Log Screen)	48
3.4	Expérimentation et discussion.....	49
	Conclusion Générale	52
	Bibliographie	54

Liste des figures

Fig. 1.1 - Le schéma général du filtrage d'informations.....	7
Fig. 1.2 - Principales classifications des systèmes de recommandations.....	9
Fig. 1.3 - Les différentes approches des systèmes de recommandations.....	11
Fig.1.4 - Principe général du filtrage collaboratif.....	14
Fig.1.5 - Représentation de la transformation d'une matrice creuse	17
Fig.1.6 - Recommandation basée sur le contenu.....	18
Fig. 2.1 - Architecture générale du système proposé.....	26
Fig. 2.2 - Un exemple d'une partie de la collection des données.....	27
Fig. 2.3 - Un exemple réel positif depuis l'ensemble de données.....	29
Fig. 2.4 - Un exemple réel négatif depuis l'ensemble de données.....	29
Fig. 2.5 - Création de la base de données.....	30
Fig. 2.6 - Diagramme de cas d'utilisation.....	33
Fig. 2.7 - Diagramme de classe.....	34
Fig. 2.8 - Diagramme de séquence « Login ».....	34
Fig. 2.9 - Diagramme de séquence « Inscription ».....	35
Fig. 2.10 - Diagramme de séquence « consulter les lieux à proximité ».....	35
Fig. 3.1 - Écran de Connexion (Login Screen).....	43
Fig. 3.2 - Écran d'Inscription (Register Screen).....	44
Fig. 3.3 - Écran du Profil Utilisateur (User Profile Screen).....	45
Fig. 3.4 - Écran des Catégories (Category Screen).....	46
Fig. 3.5 - Écran de Liste des Lieux (Places List Screen).....	47
Fig. 3.6 - Écran de Liste des Lieux et carte google map.....	48
Fig. 3.7 - Écran du Journal des Commentaires Utilisateurs.....	49

ملخص

اكتسبت أنظمة التوصية شعبية كبيرة في السنوات الأخيرة لمعالجة مشكلة فرط المعلومات من خلال تقديم توصيات مخصصة. نظام التوصية هو أداة لاسترجاع المعلومات وفرزها، مصممة لاقتراح عناصر تهم المستخدمين. تعتمد هذه الأنظمة عادة على تحليل تفضيلات المستخدم، بالإضافة إلى تقييماته الصريحة أو الضمنية للعناصر. وقد أثبتت أنظمة التوصية متعددة المعايير أنها أكثر دقة من أنظمة التوصية أحادية المعيار، لأن التقييمات متعددة المعايير تعكس تقدير المستخدم للعنصر من وجهات نظر متعددة.

لا يزال من الصعب اليوم على المستخدم عمومًا، أو السائح خصوصًا، العثور بسرعة على فندق أو مطعم أو كافيتيريا أو طبيب أو أي موقع آخر جيد بناء على موقعه الحالي. لتلبية هذه الحاجة، نقترح تطوير نظام توصية قادر على تقديم توصيات مواقع بناء على تفضيلات المستخدم، من خلال الجمع بين عدة أساليب: القائمة على المحتوى، والتصنيفية التعاونية، والتقييمات متعددة المعايير، وما إلى ذلك. يتعلم هذا النظام تصنيف المواقع ذات الصلة والتوصية بها بناء على قربها الجغرافي، وتقييمات المستخدمين وتعليقاتهم، وأعمارهم وأجناسهم، وما إلى ذلك.

في هذا العمل، تم تنفيذ مرحلة جمع تقييمات المستخدمين وملاحظاتهم على مواقع مختلفة تقع في مدينة سعيدية. ثم تم تحليل البيانات التي تم جمعها باستخدام طريقة تحليل المشاعر VADER وأخيرًا، تم تصميم وتنفيذ خوارزمية توصية للتوصية بمواقع ذات صلة مصممة خصيصًا لتفضيلات المستخدم، مع مراعاة معايير مختلفة: الفئة، والموقع، والتقييم، والتعليقات، وعمر المستخدم، وجنسه. وأخيرًا، نهدف إلى اختبار نظام التوصية الهجين متعدد المعايير الخاص بنا على هذه البيانات الواقعية لإثبات قدرته على تقديم اقتراحات ذات صلة مصممة خصيصًا لتفضيلات المستخدم.

الكلمات المفتاحية: أنظمة التوصية، المواقع، تحديد الموقع الجغرافي، التصنيفية التعاونية، تحليل المشاعر، معايير، متعددة هجينة.

Abstract

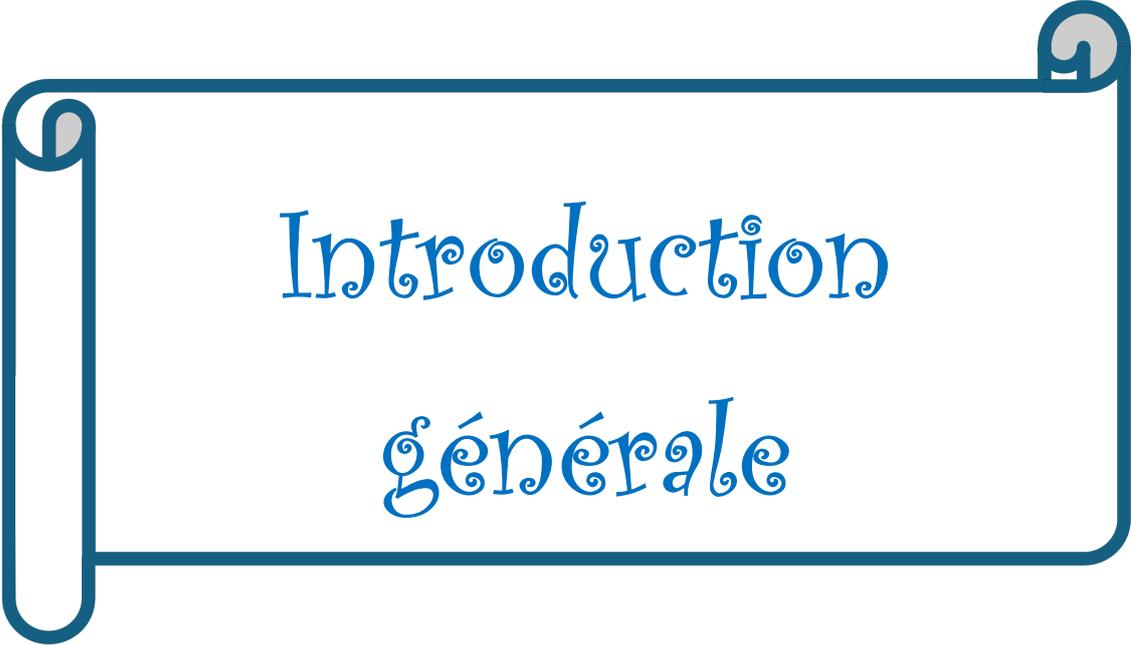
Recommender systems have become extremely popular in recent years to address the problem of information overload by providing personalized recommendations. A recommender system is an information retrieval and filtering tool designed to suggest items of interest to users. These systems typically rely on analyzing user preferences as well as their explicit or implicit evaluations of items. Multi-criteria recommender systems have proven to be more accurate than single-criteria recommender systems because multi-criteria evaluations reflect a user's appreciation of an item from multiple perspectives. Today, it is still difficult for a user in general, or a tourist in particular, to quickly find a good hotel, restaurant, cafeteria, doctor, or any other location based on their current location. To address this need, we propose developing a recommender system capable of providing location recommendations based on user preferences, by combining several methods: content-based, collaborative filtering, multi-criteria evaluations, etc. This system learns to rank and recommend relevant locations based on their geographic proximity, user reviews and comments, their ages and genders, etc. In this work, a phase of collecting user reviews and feedback was carried out on various locations located in the city of Saida. The collected data was then analyzed using the VADER sentiment analysis method. Finally, a recommendation algorithm was designed and implemented to recommend relevant locations tailored to user preferences, taking into account various parameters: category, location, rating, comments, user age, and gender. Finally, we aim to test our hybrid multi-criteria recommendation system on this real-world data to demonstrate its ability to offer relevant suggestions tailored to user preferences.

Keywords: recommendation systems, locations, geolocation, sentiment analysis, hybrid, multi-criteria.

Résumé

Les systèmes de recommandation sont devenus extrêmement répandus ces dernières années pour résoudre le problème de surcharge d'informations en fournissant des recommandations personnalisées. Un système de recommandation est un outil de recherche d'informations et de filtrage conçu pour suggérer aux utilisateurs des éléments susceptibles de les intéresser. Ces systèmes s'appuient généralement sur l'analyse des préférences des utilisateurs ainsi que sur leurs évaluations explicites ou implicites des items. Les systèmes de recommandation multicritères se sont révélés plus précis que les systèmes de recommandation à critère unique, car les évaluations multicritères reflètent l'appréciation d'un utilisateur pour un élément sous plusieurs aspects. Aujourd'hui, il est encore difficile pour un utilisateur en générale ou un touriste en particulier de trouver rapidement un bon hôtel, restaurant, cafeteria, médecin ou tous autres lieux en fonction de sa position actuelle. Pour répondre à ce besoin, nous proposons de développer un système de recommandation, capable de fournir des recommandations de lieux en fonction des préférences des utilisateurs, en combinant plusieurs méthodes : à base de contenu, filtrage collaboratif, évaluations multicritères, etc. Ce système apprend à classer et de recommander les lieux pertinents en fonction de leurs proximité géographique, des avis et des commentaires des utilisateurs, leurs âges et sexes, etc. Dans ce travail, une phase de collecte des avis et feedback des utilisateurs a été réalisée sur divers lieux situés dans la ville de Saida. Les données recueillies ont ensuite été analysées à l'aide de la méthode d'analyse de sentiments VADER. Enfin, un algorithme de recommandation a été conçu et implémenté pour recommander des lieux pertinents adaptées aux préférences des utilisateurs, en tenant compte de divers paramètres : catégorie, localisation, les évaluations (rating), les commentaires, âge et sexe des utilisateurs. Enfin, nous visons à tester notre système de recommandation hybride multicritère sur ces données réelles pour montrer qu'il parvient à proposer des suggestions pertinentes adaptées aux préférences des utilisateurs.

Mots Clés : systèmes de recommandation, lieux, géolocalisation, analyse des sentiments, hybride, multicritère.



Introduction générale

Introduction générale

Dans un monde où l'information est omniprésente et les choix de plus en plus nombreux, les systèmes de recommandation sont devenus des outils indispensables pour aider les utilisateurs à naviguer dans cette complexité et à découvrir de nouveaux contenus, produits ou services. Ces systèmes, qui s'appuient sur l'intelligence artificielle et l'apprentissage automatique, ont révolutionné la manière dont nous interagissons avec les plateformes numériques, qu'il s'agisse de plateformes de streaming, de commerce électronique ou de réseaux sociaux. Leur capacité à anticiper les préférences individuelles et à proposer des suggestions pertinentes contribue non seulement à améliorer l'expérience utilisateur, mais aussi à générer une valeur significative pour les entreprises.

Parallèlement, l'avènement des technologies mobiles et la démocratisation des smartphones ont transformé l'accès à l'information et aux services locaux. Les applications géolocalisées jouent un rôle crucial dans la découverte de lieux d'intérêt, de restaurants, d'attractions touristiques ou d'établissements commerciaux. Cependant, face à la multitude d'options disponibles, les utilisateurs peuvent se sentir dépassés, d'où la nécessité de systèmes intelligents capables de filtrer et de personnaliser les recommandations en fonction de leurs besoins spécifiques et de leur contexte géographique.

Les systèmes de recommandation utilisent principalement deux méthodes de filtrage pour proposer des recommandations personnalisées aux utilisateurs, à savoir le filtrage collaboratif et le filtrage basé sur le contenu. Le premier regroupe des techniques qui visent à opérer une sélection sur les items à présenter aux utilisateurs (filtrage) en se basant sur le comportement et les goûts exprimés de très nombreux autres utilisateurs. Le deuxième consiste à déterminer quels items coïncident le mieux avec les préférences de l'utilisateur. Le filtrage collaboratif est considéré comme la méthode la plus populaire et la plus répandue dans les systèmes de recommandation qui reposent sur les données collectées par le biais de retours d'utilisateurs, généralement sous la forme d'une matrice de notes, et tentent d'y découvrir les informations pertinentes pour caractériser et prédire les goûts des utilisateurs. Les systèmes de recommandation basés sur le filtrage collaboratif recommandent à l'utilisateur actif les items que d'autres utilisateurs ayant des préférences similaires ont aimés par le passé. Cette méthode de recommandation repose sur l'idée que, si les utilisateurs sont d'accord sur la qualité de certains items, ils seront probablement d'accord sur d'autres.

Les systèmes de recommandation multicritères se sont révélés plus précis que les systèmes de recommandation à critère unique, car les évaluations multicritères reflètent l'appréciation d'un utilisateur pour un élément sous plusieurs aspects. Aujourd'hui, il est encore difficile pour un utilisateur en générale ou un touriste en particulier de trouver rapidement un bon hôtel, restaurant, cafeteria, médecin ou tous autres lieux en fonction de sa position actuelle.

Le présent rapport s'inscrit dans cette dynamique en proposant le développement d'un système de recommandation de lieux hybride multicritères, spécifiquement conçu pour la wilaya de Saïda. L'objectif est d'offrir une solution innovante permettant aux habitants et aux visiteurs de découvrir des lieux pertinents de manière efficace et personnalisée. Pour

répondre à ce besoin, nous proposons à développer un système de recommandation, capable de fournir des recommandations de lieux en fonction des préférences des utilisateurs, en combinant plusieurs méthodes : à base de contenu, filtrage collaboratif, évaluations multicritères, etc. Ce système apprend à classer et de recommander les lieux pertinents en fonction de leurs proximité géographique, des avis et des commentaires des utilisateurs, leurs âges et sexes, etc. Dans ce travail, une phase de collecte des avis et feedback des utilisateurs a été réalisée sur divers lieux situés dans la ville de Saida. Les données recueillies ont ensuite été analysées à l'aide de la méthode d'analyse de sentiments VADER. Enfin, un algorithme de recommandation a été conçu et implémenté pour recommander des lieux pertinents adaptées aux préférences des utilisateurs, en tenant compte de divers paramètres : catégorie, localisation, les évaluations (rating), les commentaires, âge et sexe des utilisateurs. Enfin, nous visons à tester notre système de recommandation hybride multicritère sur ces données réelles pour montrer qu'il parvient à proposer des suggestions pertinentes adaptées aux préférences des utilisateurs.

Pour atteindre cet objectif, nous avons structuré notre mémoire comme suit :

Une introduction générale

Le premier chapitre posera les bases théoriques en introduisant les concepts fondamentaux des systèmes de recommandation, leurs typologies et les méthodes d'analyse de sentiment.

Le deuxième chapitre détaillera la conception de notre système de recommandation hybride multicritères, en expliquant l'architecture choisie, les algorithmes sous-jacents et les choix technologiques.

Le troisième chapitre sera consacré à l'implémentation de l'application "Saïda Finder", en présentant les outils et les technologies utilisées, les étapes de développement, ainsi que les résultats obtenus et les perspectives d'amélioration future.

Enfin, une conclusion générale qui résume notre travail.



Chapitre 1 :

Introduction aux Systèmes de
Recommandation

Chapitre 1 : Introduction aux Systèmes de Recommandation

1.1 Introduction

Le développement de l'internet nous a permis d'être au milieu d'une surcharge d'informations. Par exemple, une personne qui souhaite lire un cours se retrouve confronté à un grand nombre de propositions de cours. Cela rend le choix d'un cours très difficile. Des systèmes de recommandation ont émergé pour résoudre ce problème. Nous commençons ce chapitre en définissant ce qu'est un système de recommandation. Ensuite, nous présenterons les trois approches de filtrage qui permettent la recommandation. Nous décrirons et démontrerons les nombreuses mesures de similarité qui permettent aux systèmes de recommandation d'établir des liens entre les concepts. Enfin, nous mentionnerons quelques travaux significatifs dans le domaine ainsi que les limitations et les inconvénients des systèmes de recommandation.

Ce chapitre propose une exploration des concepts fondamentaux relatifs aux systèmes de recommandation, en présentant leurs différentes catégories et leurs champs d'application. Nous examinerons également, de manière générale, les approches d'analyse des sentiments, en mettant l'accent sur la méthode VADER, que nous avons intégrée dans le cadre de cette étude. Une revue des travaux existants est présentée en fin de chapitre.

1.2. Historique

Dans les premiers jours de l'informatique, les systèmes de recommandation étaient largement reconnus. Le "Système de Lentille d'Information"[27] est considéré comme le premier système de recommandation de ce type. À l'époque, la liste de distribution basée sur les groupes d'intérêt était la solution la plus populaire pour le partage d'informations dans un environnement de messagerie électronique.

La première définition du filtrage a également été donnée par Malone : "Bien que le terme ait une connotation littérale de laisser des choses de côté (filtrage négatif: suppression), nous l'utilisons ici dans un sens plus général, qui consiste à sélectionner des éléments dans un ensemble plus large de possibilités (filtrage positif: sélection)".

Le terme "filtrage collaboratif" par le système "Tapestry" a été introduit dans la littérature académique [27]. Le système de recommandation a été créé en 1992 par le centre de recherche "Xerox" aux États-Unis et a été intégré à une application de messagerie électronique pour permettre aux utilisateurs de générer des recherches permanentes basées sur des annotations utilisateur (tags).

Quelques années plus tard, en 1994 et 1995, divers systèmes de recommandation académiques ont été créés, notamment le système de recommandation musicale Ringo et le système de recommandation de nouvelles et de films créés par Group Lens [27]. Le

filtrage collaboratif de livres (HAL thesis), de films, d'émissions de télévision, de pages web, d'articles et de liens Internet constitue également la base de ces deux plateformes.

Ensuite, afin d'améliorer sa position en tant que service de recommandation de films, Netflix a introduit le Prix Netflix en 2006. En conséquence, il propose désormais plus de 17 000 films dans sa bibliothèque. De nos jours, les systèmes de recommandation sont largement utilisés dans un large éventail d'applications web.

Les systèmes de recommandation ont ensuite gagné en popularité dans d'autres domaines d'application grâce au développement de l'Internet et des applications web. Nous pouvons citer :

- Les systèmes de recommandation de films, tels que Mobile et Eachmovie.
- Les systèmes de recommandation de livres (Bookcrossing).
- Les systèmes de recommandation musicale (LastFM6).
- Les systèmes de recommandation d'articles de presse.
- Les systèmes de recommandation de blagues.
- Les systèmes de recommandation introduits sur les sites de commerce électronique (Amazon).
- Les systèmes de recommandation de restaurants.
- Les systèmes de recommandation intégrés dans les extranets documentaires (Extranet documentaire du Crédit Agricole).
- Les systèmes de recommandation intégrés aux moteurs de recherche (le moteur de recherche AOL).
- Les systèmes de recommandation mis en œuvre sur les sites de recrutement (JobFinder).
- Les systèmes de recommandation bibliographique.

Dans chaque système de recommandation créé jusqu'à présent, la collecte de données sur les utilisateurs et/ou les objets est une étape cruciale du processus de personnalisation. Les types de données que les systèmes de recommandation peuvent utiliser et les problèmes qui surviennent lors de la collecte de données sont détaillés dans la section suivante.

1.3 Définition des systèmes de recommandation

Étant donné la variété des classifications proposées pour les systèmes de recommandation, il existe de nombreuses façons de les définir ; cependant, la définition générale de Robin Burke définit les systèmes de recommandation comme suit :

"Des systèmes capables de fournir des recommandations personnalisées pour guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un grand espace de données."

Un système de recommandation, parfois appelé système de filtrage d'informations, est un système qui filtre les données entrantes en fonction des besoins uniques de chaque acteur. En d'autres termes, un système de filtrage rassemble, choisit, catégorise et recommande à l'utilisateur des informations susceptibles de correspondre à ses intérêts à long terme afin de personnaliser sa recherche d'informations dans un certain domaine d'intérêt.

L'utilisateur et l'élément sont les deux entités fondamentales que l'on trouve dans tous les systèmes de recommandation. L'individu qui utilise un système de recommandation, fournit des retours d'informations sur différents produits et reçoit de nouvelles recommandations du système est appelé "utilisateur". Le nom de base de ce que le système suggère aux utilisateurs est un élément.

Les données d'entrée d'un système de recommandation sont déterminées par le type d'algorithme de filtrage utilisé. Ils tombent généralement dans l'un des groupes suivants :

- Les estimations, également appelées votes, transmettent les opinions des utilisateurs sur les articles (par exemple, de 1 très mauvais à 5 excellent).
- Les données démographiques : comprennent des détails sur le pays de l'utilisateur, son niveau d'éducation, son âge et son sexe. Ce type d'information est généralement spécifiquement collecté et est souvent difficile d'accès.
- Les données de contenu sont issues d'une analyse textuelle des documents relatifs aux éléments évalués par l'utilisateur. Pour déterminer un profil utilisateur, les caractéristiques qui ont été extraites de l'analyse sont entrées dans le processus de filtrage.

Le système de recommandation calcule les recommandations en utilisant des profils qui indiquent des préférences généralement stables des utilisateurs pour effectuer le filtrage. Pour cela, il prédit les scores que l'utilisateur est susceptible d'attribuer au contenu. Le système de recommandation utilise les retours d'information de l'utilisateur sur la pertinence du contenu (documents) qu'il reçoit pour ajuster ce profil au fil du temps. La fonction de jugement du système, par exemple, analyse le flux de documents entrants dans la figure 1.1 pour recommander les documents que l'utilisateur choisit en fonction de son profil.

Pour aider le système à mieux comprendre les besoins d'information de l'utilisateur et à fournir des recommandations plus pertinentes, l'utilisateur doit également évaluer régulièrement les recommandations.

Les trois parties suivantes constituent un système de recommandation :

- Les producteurs : Ce sont ceux qui feront des recommandations, ils "fourniront" les données aux utilisateurs.
- Le module de calcul : C'est l'algorithme lui-même. À l'entrée se trouvent toutes les données et la demande et à la sortie les différentes recommandations.
- Le consommateur : C'est celui qui demande la recommandation.

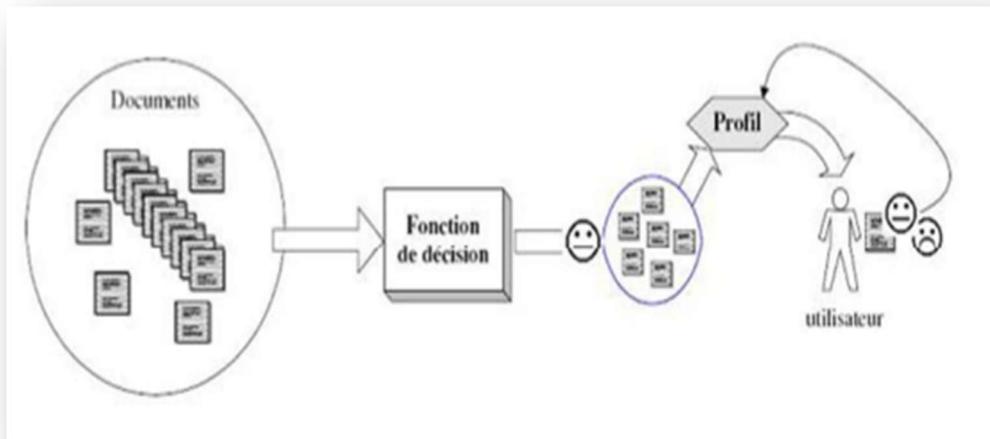


Fig. 1.1 : Le schéma général du filtrage d'informations [30].

1.4 Objectifs des systèmes de recommandation

Un système de recommandation cherche à offrir des ressources pertinentes à un utilisateur selon ses choix. Ainsi, ce dernier raccourcit son temps de recherche mais reçoit également des recommandations du système qu'il n'aurait pas pensé à rechercher par lui-même.

Des systèmes comme ceux de l'industrie du commerce électronique ont été établis en partie en raison de la croissance et de la popularité du Web. Par exemple, les sites de commerce électronique bien connus tels qu'Amazon ainsi que le moteur de recherche d'articles de référence CiteSeer.

Au début, les systèmes de recommandation peuvent être envisagés comme une solution pour les personnes qui ont du mal à décider lorsqu'elles utilisent un moteur de recherche d'informations "classique".

La base de la recherche d'informations est l'idée d'indexer les données pour faciliter les demandes des utilisateurs. La recherche documentaire, en particulier, relève du domaine de la recherche d'informations et consiste à interroger une base documentaire à l'aide de requêtes en langage naturel ou de recherches par mots-clés (également appelées requêtes ad hoc).

Les moteurs de recherche Web, tels que Google et Yahoo!, sont des exemples bien connus d'outils qui utilisent les principes de ce domaine. Les utilisateurs saisissent leur requête de recherche (ensemble de mots-clés) pour trouver les informations qu'ils recherchent. L'index de tous les documents de la base de données du moteur de recherche est ensuite croisé avec ces mots-clés. L'utilisateur se voit alors présenter un groupe de résultats quelque peu liés à sa requête initiale.

La surabondance d'informations que l'utilisateur doit trier est la difficulté fondamentale de ce type de technologie (résultats de recherche d'informations). Par exemple, la plupart du temps, les utilisateurs arrêtent de consulter des informations potentiellement pertinentes et se contentent de parcourir la première page des résultats des moteurs de recherche, en ignorant les autres. Les recommandations n'ont aucun impact sur cette situation, l'un des objectifs des systèmes de recommandation est de parcourir de manière transparente cette grande quantité de données pour l'utilisateur.

En ce qui concerne les recommandations, la création de profils d'utilisateurs est la première étape dans la recherche documentaire. En fait, comme le conseille spécifiquement Google, l'utilisateur est désormais perçu pour évaluer les résultats de sa requête à la lumière de ses recherches passées et des pages qu'il a visitées. C'est un système qui reconnaît d'abord l'utilisateur avant d'examiner son comportement. Cette méthode, également connue sous le nom d'analyse de traces, constitue l'un des principaux piliers du filtrage d'informations qui sous-tend les systèmes de recommandation.

Cela établit une distinction entre les systèmes de recommandation, où la participation de l'utilisateur n'est pas spécifiquement induite par les moteurs de surveillance, et les systèmes de recherche d'informations, où l'utilisateur demande expressément des conseils et de l'aide pour prendre des décisions appropriées.

1.5 Classification des systèmes de recommandation

De nombreuses stratégies et techniques ont été proposées pour classer les systèmes de recommandation. Plusieurs termes sont parfois utilisés pour désigner la même stratégie ou méthodologie. Pour cette raison, plusieurs chercheurs ont proposé une nomenclature unifiée, comme on peut le voir dans la Figure 1.2 et ont exprimé un intérêt pour la classification de différentes figures. Les catégories suivantes sont répertoriées :

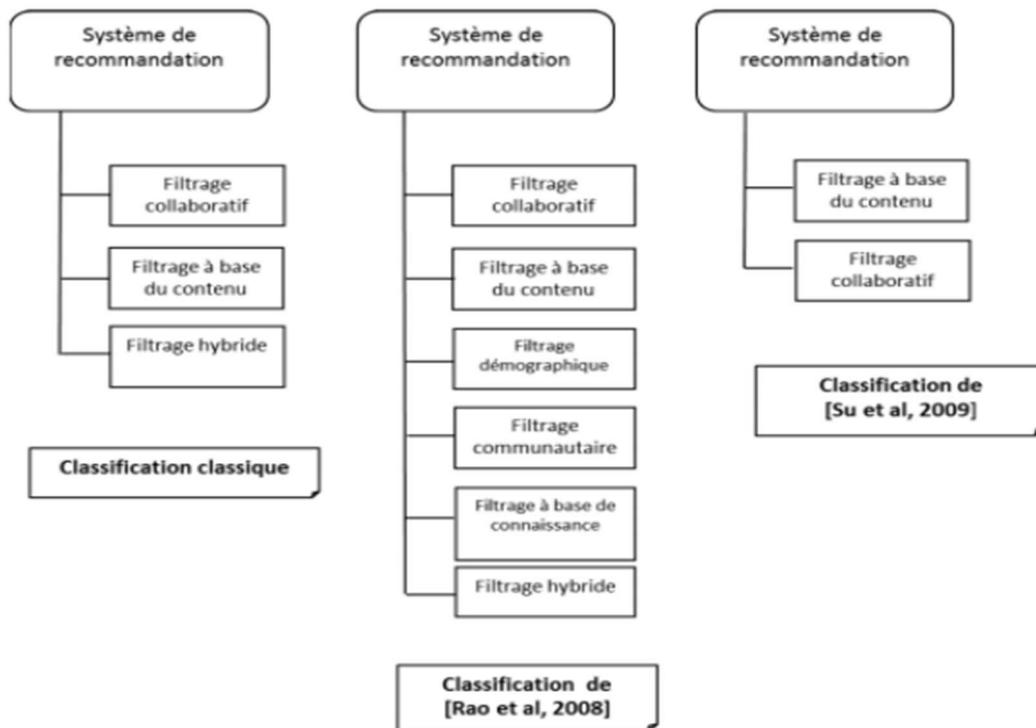


Fig. 1.2 : Principales classifications des systèmes de recommandations [30].

1.5.1 Classification classique

Cette classification de [24], a créé un voyage extrêmement fascinant pour recueillir les nombreux points de vue. De plus, ce dernier forme la base d'une quantité croissante de travail. Trois catégories de filtrage ont été distinguées : le filtrage basé sur le contenu, le filtrage hybride et le filtrage collaboratif.

1.5.2 La classification de Su et al., 2009

Dans les systèmes coopératifs, c'est une classification. La classification des approches hybrides en méthodes de collaboration hybride est proposée par les auteurs comme une sous-classification. Le filtrage collaboratif est divisé en trois groupes par Su et al. [29]:

- Techniques de filtrage collaboratif basées sur la mémoire : pour les k voisins les plus proches.
- Méthodes de filtrage collaboratif basées sur un modèle comprenant une gamme de méthodes telles que la factorisation de matrices, le regroupement, les réseaux bayésiens et les processus de prise de décision de Markov.
- Filtrage collaboratif hybride : une méthodologie qui mélange une ou plusieurs autres approches avec une méthode de suggestion de filtrage collaboratif.

1.5.3 La classification de Rao et Talwar, 2008

Cette classification repose sur la source d'information consultée. Rao et Talwar ont fourni une liste complète de systèmes de recommandation pour de nombreux domaines d'application, tant dans des environnements académiques qu'industriels. Basés sur les données utilisées, ils peuvent être largement divisés en six catégories :

- le filtrage coopératif
- le filtrage basé sur le contenu
- le filtrage hybride
- le filtrage démographique
- le filtrage basé sur la communauté
- le filtrage basé sur les connaissances.

1.6 Etude des algorithmes de recommandations

L'étude des algorithmes de recommandations a apporté des centaines d'algorithmes qui peuvent être classés sur base de leurs approches. C'est sur cette base hiérarchique que nous allons explorer les algorithmes les plus connus. Dans un premier temps, nous allons explorer le filtrage basé sur le contenu qui présente les algorithmes les plus simples. Nous étudierons ensuite l'approche basée sur le filtrage collaboratif qui représente la partie la plus importante de par ses performances et sa large adoption. Enfin, j'expliquerai brièvement l'approche hybride qui consiste à mélanger les recommandations des deux approches précédentes dans le but de pallier certains défauts.

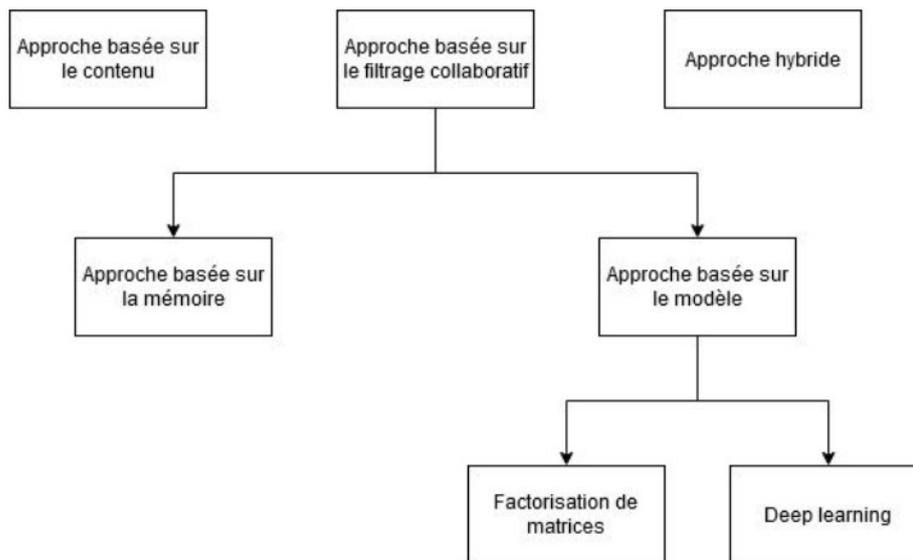


Fig. 1.3 : Les différentes approches des systèmes de recommandations.

1.6.1 Recommandation basée sur le filtrage collaboratif

Le filtrage collaboratif (Collaborative Filtering « CF ») a pour principe d'exploiter les évaluations faites par des utilisateurs sur certains documents (contenus), afin de recommander ces mêmes documents à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le contenu des documents.

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grâce à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo.

De plus, l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».

Un autre avantage du filtrage collaboratif est que les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des documents tels que la diversité, la nouveauté, etc.

Le CF souffre de plusieurs gros problèmes. Le problème principal étant le démarrage à froid : c'est le fait qu'un utilisateur doit voter sur beaucoup d'objet avant d'obtenir les recommandations.

1.6.1.1 Processus du filtrage collaboratif

Le processus du filtrage collaboratif suit les étapes données ci-dessous :

1.6.1.1.1 Evaluation des recommandations

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés. Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit :

- **Explicite** : L'utilisateur donne une valeur numérique sur une échelle donnée (par exemple de 1 à 5, ou de 1 à 10, etc.), ou bien, une valeur qualitative de satisfaction, par exemple, mauvaise, moyenne, bonne et excellente.
- **Implicite** : Le système induit la satisfaction de l'utilisateur à travers ses actions. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation très mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation.

1.6.1.1.2 Formation des communautés

Le processus de formation des communautés est le noyau d'un système de filtrage collaboratif. Pour chaque utilisateur, le système doit calculer sa communauté, généralement cela se fait par la proximité des évaluations des utilisateurs. Pour ce faire, on peut calculer, dans un premier temps, la proximité entre un utilisateur donné et tous les autres. Ensuite, et afin de créer contrairement la communauté de l'utilisateur, on applique la méthode des voisins les plus proche en utilisant un seuil pour le niveau de proximité ou un seuil pour la taille maximale de la communauté, en raison de sa performance et sa précision.

1.6.1.1.3 Production des recommandations

Dans ce dernier processus, une fois la communauté de l'utilisateur créée, le système prédit l'intérêt qu'un document particulier peut présenter pour l'utilisateur en s'appuyant sur les évaluations que les membres de la communauté ont faites sur ce même document. Lorsque l'intérêt prédit dépasse un certain seuil, le système recommande le document à l'utilisateur.

1.6.1.1.4 Profils et communautés

Ici, nous discutons les profils basés sur l'historique des évaluations des utilisateurs, ainsi que les communautés, qui sont les deux facteurs clés d'SFC. Le problème de la surcharge d'information peut être pallié par la personnalisation de l'accès aux informations, en utilisant des profils représentant des intérêts relativement stables des utilisateurs. En d'autres termes les profils des utilisateurs sont utilisés comme des critères persistant dans la recherche d'information.

a-) Profil de l'utilisateur :

Le profil utilisateur est composé de prédicats pondérés. Le poids d'un prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un nombre réel compris entre 0 et 1. Le profil s'enrichit progressivement au fur et à mesure que l'utilisateur évalue des documents reçus. Outre les informations d'identification de base (par exemple, l'identifiant ou des éléments d'état civil), le profil de l'utilisateur peut regrouper des informations très diverses selon les besoins.

Parmi celles-ci, nous pouvons citer :

- Des caractéristiques personnelles pouvant influencer fortement l'interaction (âge, sexe,)
- Les intérêts et les préférences générales de l'utilisateur relatives à la tâche à accomplir, qui permettent une adaptation à ses attentes.
- Qualité. Cette dimension contient tous les facteurs reflétant les préférences relatives à la qualité de l'information, comme la disponibilité de données, la concision, le style et la structure du document, etc. Dans cette dimension, nous nous intéressons en particulier à la diversité de l'information.
- Sécurité. La dimension de sécurité dans le contexte du filtrage collaboratif, est le niveau de confidentialité concernant tous les autres critères.
- Un historique des interactions avec le service, qui peuvent permettre de modéliser les habitudes comportementales.

b-) Communautés

La notion de communauté dans un système de filtrage collaboratif est définie comme le regroupement des utilisateurs en fonction de l'historique de leurs évaluations, afin que le système calcule des recommandations. Selon cette optique, les profils sont un facteur interactif, alors que les communautés sont considérées comme un facteur interne du système.

1.6.1.1.5 Exemple

Dans la figure 1.4, nous schématisons le principe du filtrage collaboratif. On suppose que l'on a des communautés formées par la proximité des évaluations des utilisateurs. Le document d sera recommandé à l'utilisateur u , car ce document est apprécié par la communauté G où se trouve l'utilisateur.

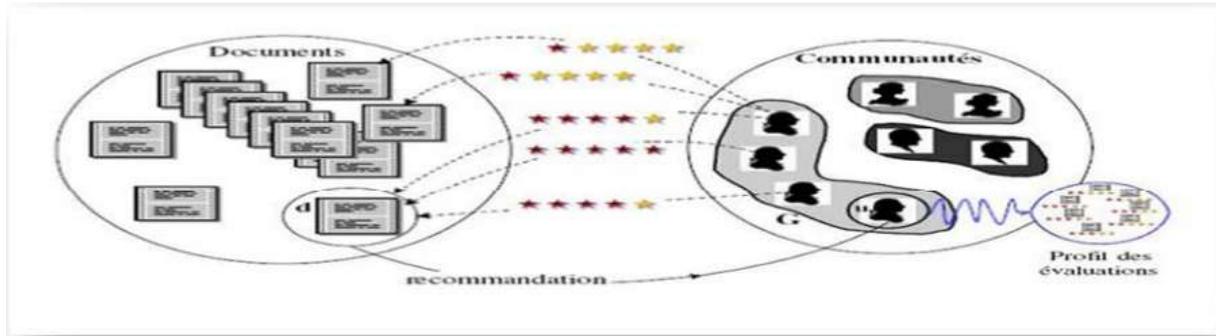


Fig.1.4 : Principe général du filtrage collaboratif

1.6.1.2 Approche basée sur la mémoire

L'approche basée sur la mémoire utilise les interactions passées des utilisateurs pour calculer les similitudes entre ceux-ci ou entre les items. Pour trouver la note r qu'un utilisateur u donnerait à un item i , l'approche recherche les utilisateurs similaires à u qui ont noté l'item i et calcule la note r en fonction des notes des utilisateurs trouvés à l'étape précédente. Afin de trouver les U utilisateurs les plus similaires à l'utilisateur u , on calcule la similarité sur base des items communément notés avec l'utilisateur comparé en calculant leur distance ou leur similarité.[26]

NB : Notez que deux utilisateurs A et B peuvent être considérés comme absolument similaires dans la métrique de similarité cosinus malgré des évaluations différentes. Un exemple serait un utilisateur critique de cinéma qui attribue toujours des notes inférieures à la moyenne, mais dont le classement des éléments de sa liste serait similaire à celui des évaluateurs moyens comme B . Pour tenir compte de ces préférences individuelles des utilisateurs, il faut amener tous les utilisateurs au même niveau en supprimant leurs préjugés. Ceci peut se faire en soustrayant la note moyenne donnée par cet utilisateur à tous les items de chaque item noté par cet utilisateur.

Après avoir déterminé une liste d'utilisateurs similaire à un utilisateur u , on calcule la note r que u donnerait à un certain item i .

On considère que la note r d'un utilisateur pour un item i sera proche de la moyenne des notes attribuées à i par les U utilisateurs les plus similaires à u . La formule mathématique de la note moyenne donnée par U utilisateurs se calcule avec formule dont la version la plus simple est :

$$\hat{r}_{ui} = \frac{1}{n} * \sum_{u' \in U} r_{u',i} [1.1]$$

r : représente la note donnée par un utilisateur u à un item i et U représente le groupe d'utilisateurs similaires à u .

Il est également possible de multiplier la note par le degré de similarité entre les deux utilisateurs afin de donner plus de poids aux notes d'utilisateurs fort similaires à u :

$$\hat{r}_{ui} = k * \sum_{u' \in U} sim(u, u') * r_{u',i} [1.2]$$

k : est un facteur de normalisation.

Enfin, on peut également prendre en compte les notes moyennes de l'utilisateur u dans le calcul étant donné que les utilisateurs peuvent avoir tendance à voter différemment les uns des autres :

$$\hat{r}_{ui} = \bar{r}_u + k * \sum_{u' \in U} sim(u, u') * (r_{u',i} - \bar{r}_{u'}) [1.3]$$

\bar{r}_u : est la moyenne des notes de l'utilisateur u pour tous les items notés par u .

Différence entre l'approche basée sur les utilisateurs et les items

Il existe deux approches pouvant être utilisées pour trouver des recommandations avec le filtrage collaboratif. Ces deux approches sont mathématiquement assez similaires, mais il existe une différence conceptuelle entre les deux. Voici comment elles se comparent :

- Basé sur les utilisateurs : pour un utilisateur u , le vote pour un item i , qui n'a pas encore été voté par l'utilisateur u est trouvé en prenant les U utilisateurs les plus similaires qui ont noté l'item i et en calculant le vote basé sur les votes de ces U utilisateurs.
- Basé sur les items : pour un item i , le vote par un utilisateur u , qui ne l'a pas encore voté est trouvé en prenant les I items les plus similaires qui ont été notés par l'utilisateur u en calculant le vote basé sur les votes de ces I items

Dans un système où il y a plus d'utilisateurs que d'items, le filtrage basé sur les items est plus rapide et plus stable que celui basé sur les utilisateurs. Il est efficace car en général, la note moyenne reçue par un item ne change pas aussi rapidement que la note moyenne attribuée par un utilisateur à différents éléments. Il est également connu pour être plus performant que l'approche basée sur les utilisateurs lorsque la matrice de notation est fortement éparse.[26]

1.6.1.3 Filtrage collaboratif basé sur un modèle

Comme son nom l'indique, ce type d'algorithme repose sur des modèles afin de minimiser la complexité. Ces modèles estiment ou apprennent un modèle qui est ensuite utilisé pour les prédictions en utilisant la base de données des notes des utilisateurs. Pour simplifier la complexité, ils peuvent être construits sur des classificateurs pour produire des classes.

1.6.1.3.1 La factorisation de matrices

La factorisation de matrices, ou décomposition de matrices est une méthode qui permet d'accélérer la recherche de recommandations. L'idée derrière la factorisation matricielle est de représenter les utilisateurs et les items dans un espace latent de dimension inférieure à celui de base en décomposant la matrice initiale en plusieurs autres matrices. Cette

réduction de dimensionnalité permet de faire face aux problèmes de scalabilité du traitement de ces matrices qui peuvent se révéler volumineuses et très éparées. Pour retrouver la matrice originale, il suffira de faire le produit de ces matrices entre elles. Cette décomposition d'une matrice creuse en deux matrices denses de dimensions inférieures nous permet d'économiser du stockage et d'accélérer les calculs. Ces avantages en ont fait une méthode très utilisée dans le domaine du filtrage collaboratif.[26]

1.6.1.3.2 Le modèle des facteurs latents

Le modèle des facteurs latents représente les items et les utilisateurs par des vecteurs de caractéristiques de même taille, les facteurs latents. Plus la correspondance entre les facteurs latents d'un utilisateur et d'un item est grande, plus il y a de chance pour que le film corresponde aux goûts de l'utilisateur. Bien que ces facteurs latents représentent des caractéristiques, il ne faut pas faire l'erreur de vouloir mettre une étiquette dessus car nous ne pouvons faire que des suppositions sur leurs significations.

L'objectif principal des facteurs latents est d'approximer au mieux la matrice des relations user-item. L'estimation de cette matrice R est égale à la multiplication de la matrice des \hat{R} facteurs latents des utilisateurs P par la matrice transposée des facteurs latents des items Q .[26]

$$R \approx P * Q^T = R [1.4]$$

Approximation de la matrice R par la multiplication de la matrice P et Q transposée.

Pour trouver la correspondance entre un utilisateur et un film, on multiplie leurs facteurs latents :

$$\hat{R}_{ui} = P_u^T * q_i = \sum_{k=1}^k p_{uk} * q_{ki} [1.5]$$

Aproximation d'une relation user-item par la multiplication des facteurs latents de l'utilisateur u et de l'item i . k représente le nombre de facteurs latents.

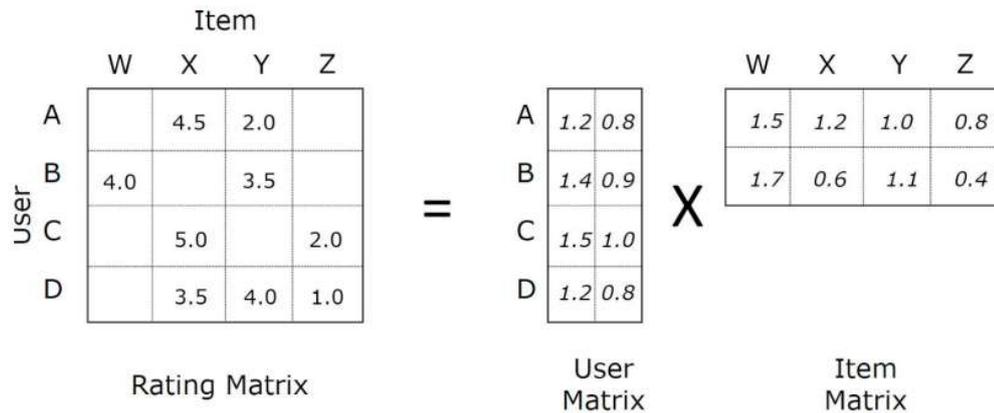


Fig.1.5 : Représentation de la transformation d’une matrice creuse en matrices denses à 2 facteurs latents

Méthodes d’apprentissage

Afin de trouver les facteurs latents qui estiment le mieux la matrice \hat{R} , les matrices Q et P sont d’abord initialisées avec des valeurs aléatoires. Ensuite, on minimise l’erreur quadratique entre la relation estimée et la réelle grâce la formule :

$$\min_{q_i, p_u} \sum_{u,i \in k} (r_{ui} - \hat{r}_{ui})^2 + \lambda (\|q_i\|^2 + \|p_u\|^2) \quad [1.6]$$

Formule de la minimisation de la différence entre la relation estimée et celle contenue dans la matrice originelle. Le deuxième terme est une régularisation qui permet de ne pas surestimer les données observées.

Afin de minimiser cette erreur, nous devons passer par un algorithme d’apprentissage automatique qui va progressivement diminuer l’erreur des estimations. Les deux techniques les plus utilisées sont l’apprentissage par descente de gradient stochastique et l’apprentissage par les moindres carrés alternés.

1.6.2. Recommandation basée sur le contenu

Les évaluations des utilisateurs d'une collection de documents ou d'éléments servent de base aux systèmes de recommandation basés sur le contenu. L'objectif suivant est de comprendre la logique derrière votre décision de considérer un certain élément comme important ou non.

On peut le considérer comme un système de recherche d'informations exploitant le profil de l'utilisateur.

Les systèmes de recommandation basés sur le contenu présentent l'avantage de vous permettre de lier des produits au profil de l'utilisateur, notamment en appliquant des méthodes d'intelligence artificielle. L'utilisateur est autonome et peut recevoir des recommandations même lorsqu'il est le seul à utiliser l'application. Cependant, ce type de système présente plusieurs inconvénients :

- L'effet de "tunnel" : l'utilisateur ne peut pas avoir une large gamme de sujets car ses besoins deviennent de plus en plus spécialisés. Pire encore, puisqu'il n'est pas explicitement inclus dans le profil de l'utilisateur, un nouvel axe de recherche dans une certaine région pourrait ne pas être pris en compte.
- Problème de démarrage à froid : Un utilisateur novice du système trouve difficile de décrire ses intérêts et de créer un profil.

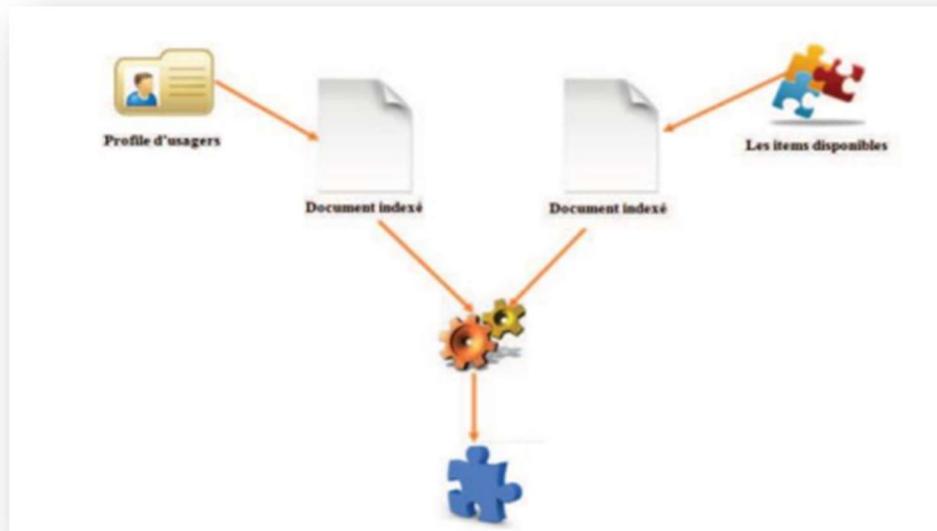


Fig.1.6 : Recommandation basée sur le contenu [25]

Il existe deux types de recommandations basées sur le contenu : basées sur des mots-clés et basées sur la sémantique :

1.6.2.1- Recommandation basée sur les mots-clés

On peut utiliser l'approche de recommandation basée sur le contenu pour suggérer des sites web, des films, des articles de presse, des restaurants, et plus encore. En prenant comme exemple un système de recommandation d'articles scientifiques basé sur le contenu, le système suggérera des publications liées à la génétique à un utilisateur qui lit souvent des articles sur ce sujet. En effet, des termes comme "ADN", "gène" et "protéine" sont fréquemment utilisés dans ces textes.

1.6.2.2- Recommandation basée sur la sémantique

De nombreuses techniques ont été utilisées pour incorporer la sémantique dans le processus de recommandation. Plusieurs facteurs sont pris en considération lors de l'adressage de ces méthodes :

- Le type de source de connaissances impliqué (lexique, ontologie, etc.).
- Les techniques adoptées pour l'annotation ou la représentation des éléments.

- Le type de contenu inclus dans le profil utilisateur.
- La stratégie de correspondance entre les éléments et le profil.

Les systèmes de recommandation basés sur la sémantique se développent en parallèle avec les techniques et les ressources disponibles dans l'espace du Web sémantique.

1.6.3 Filtrage hybride

Lorsque deux ou plusieurs systèmes de recommandation distincts sont combinés, on parle de système hybride. Les recommandations collaboratives et les recommandations basées sur le contenu sont souvent considérées comme complémentaires.

En général, l'hybridation se fait en deux étapes pour résoudre les lacunes de chaque approche lorsqu'elle est utilisée seule et pour capitaliser sur leurs forces :

- Appliquer séparément le filtrage collaboratif et d'autres techniques de filtrage pour produire des recommandations potentielles.
- Combinaison de ces ensembles de recommandations préliminaires en utilisant les méthodes suivantes : combinaison et amélioration des caractéristiques et du niveau moyen , pondération, mélange, cascading, commutation, etc. afin de fournir aux utilisateurs les recommandations finales.

En général, les systèmes hybrides gèrent les profils d'utilisateurs orientés contenu, et la comparaison de ces profils conduit à la création de communautés d'utilisateurs qui permettent le filtrage collaboratif

1.7 Autres classifications

Les typologies précédentes ont servi de base à l'identification d'autres sous-catégories de systèmes de recommandation, comme les systèmes basés sur la connaissance (KBRS), les ontologies (OBRS), les systèmes intelligents (IRS), basés sur la démographie (DBRS), sensibles au contexte (CARS), sensibles au temps (TiARS), et basés sur les avis (RBRS).

1.7.1 Systèmes de recommandation basés sur la connaissance (KBRS)

Ces systèmes opèrent en exploitant la connaissance du domaine d'application et les informations relatives aux intérêts spécifiques de l'utilisateur [16]. Ils intègrent trois types de connaissances : sur les Items, sur les Utilisateurs, et sur la similarité entre utilisateurs et Items. Les approches KBRS sont particulièrement adaptées aux domaines complexes où les Items ne sont pas fréquemment acquis.

Leur principal avantage réside dans l'absence de problème de démarrage à froid (cold start). En revanche, leur limitation majeure est la nécessité de mobiliser une expertise significative en ingénierie des connaissances [14].

1.7.2 Systèmes de recommandation basés sur les ontologies (OBRS)

Les OBRS [Ontology-Based Recommendation Systems] s'appuient sur une compréhension approfondie du profil utilisateur, des Items et du domaine concerné. Dans un OBRS, l'ontologie y joue un rôle central pour la représentation des connaissances [7].

Étroitement liés aux KBRS (dont ils utilisent l'ontologie pour la représentation), les OBRS facilitent la gestion de certaines limitations des RS traditionnels, telles que la rareté des données [12], le démarrage (ramp-up) et la sur-spécialisation. Ils dépendent cependant fortement de l'évaluation de l'adéquation au domaine.

Cette spécificité rend les OBRS particulièrement pertinents pour l'e-commerce, l'e-learning et le tourisme. Néanmoins, le développement d'ontologies adaptées représente un processus complexe et coûteux [5].

1.7.3 Systèmes de recommandation intelligents (IRS)

Les IRS [Intelligent Recommendation Systems] constituent une évolution des RS basés sur la connaissance. Ils intègrent des capacités d'exploration des connaissances, d'apprentissage, d'identification de nouvelles tendances à partir des avis, et de détection des préférences. Un IRS se caractérise par les composants suivants :

1. Méthodes d'apprentissage.
2. Modèles de représentation des connaissances.
3. Mécanismes de raisonnement.

De plus, un IRS peut mobiliser cinq types de modèles de connaissance dans un contexte donné : relatifs aux Items, aux Utilisateurs, au Contexte, aux Commentaires, et aux Domaines. Un IRS utilise les informations sur les utilisateurs via un mécanisme d'apprentissage pour optimiser ses performances [5].

1.7.4 Systèmes de recommandation basés sur la démographie (DBRS)

Le fonctionnement des DBRS [Demographic-Based Recommendation Systems] repose sur l'identification d'utilisateurs partageant des caractéristiques démographiques similaires à celles de l'utilisateur cible, afin de prédire les évaluations potentielles des Items. Plus précisément, un DBRS exploite les données démographiques des utilisateurs. L'objectif principal est de segmenter les utilisateurs en fonction de leurs attributs et données démographiques (sexe, âge, localisation, etc.) stockés dans leur profil. Contrairement aux systèmes collaboratifs (CRS) et basés sur le contenu (CBRS), la méthode démographique n'exige pas d'historique d'évaluations préalables de la part des utilisateurs [5].

1.7.5 Systèmes de recommandation sensibles au contexte (CARS)

Les CARS [Context-Aware Recommendation Systems] intègrent des informations contextuelles pour affiner et améliorer la pertinence des recommandations [5]. Les

préférences d'un utilisateur peuvent varier considérablement en fonction de l'environnement immédiat (humeur, saison, moment de la journée, lieu, compagnie, occasion, etc.). Par exemple : le choix d'un film (lieu, heure, compagnie, etc.), un voyage (destination, heure, météo, conditions de circulation, etc.), ou l'écoute de musique (heure, lieu, humeur, occasion, etc.) [17]. Ces systèmes peuvent être eux-mêmes classés en :

a- Systèmes de recommandation sensibles au temps (TiARS) :

Ces systèmes [TiARS] mettent un accent particulier sur le facteur temporel en tant qu'information contextuelle. Le profil et les intérêts des utilisateurs peuvent évoluer au fil du temps, et cette dynamique est prise en compte dans ce type de système de recommandation [5].

b- Systèmes de recommandation basés sur les avis (RBRS) :

Les RBRS [Review-Based Recommendation Systems] sont conçus pour atténuer, voire éliminer, les problèmes de démarrage à froid et de rareté des données. Un RBRS opère en analysant les profils des produits et des utilisateurs à partir des avis textuels. Pour cela, il examine et traite les avis à l'aide de techniques d'exploration de texte, incluant la recherche d'opinions et l'analyse de texte. La recommandation d'Items se fait de manière similaire à celle des CBRS et CRSs [5].

1.8 La recommandation et l'analyse des sentiments

L'analyse des sentiments (également appelée opinion mining ou sentiment analysis) fait partie intégrante du traitement automatique du langage (TAL). Elle a pour but d'identifier et de qualifier le sentiment, l'opinion ou l'évaluation exprimée (positive, négative ou neutre) au sein d'une unité textuelle (mot, phrase, paragraphe ou document) [10]. On peut la décrire comme le processus par lequel un système détermine si un texte exprime une opinion positive, négative ou neutre, en se basant sur le contenu.

1.8.1 Approches pour mesurer le sentiment

Pour mesurer le sentiment dans les textes, on distingue principalement deux approches : l'approche lexicale et l'approche statistique ou par apprentissage automatique. Une troisième approche, hybride, combine des éléments linguistiques et des modules de classification.

1.8.2 L'approche lexicale

L'approche lexicale repose sur l'utilisation de lexiques préconstruits. Dans ces lexiques, chaque terme est associé à une polarité (généralement positif, négatif, et dans certains cas, neutre). Il s'agit principalement de verbes et d'adjectifs, mais aussi de noms et d'adverbes. Cette méthode offre une bonne transparence des critères de classification. L'ajout, la modification ou la suppression de termes dans les lexiques sont simples et permettent

d'améliorer les résultats. Cependant, la construction manuelle de tels lexiques est un travail laborieux et leur couverture reste relativement limitée. Les termes doivent être explicitement listés pour être détectés. De plus, les lexiques d'analyse des sentiments se limitent souvent à des termes uniques (unigrammes) [10].

1.8.3 L'approche d'apprentissage automatique

L'apprentissage automatique (Machine Learning) est un sous-domaine de l'intelligence artificielle qui confère aux systèmes une capacité d'interprétation et de décision grâce à des algorithmes. Le principe est de permettre aux ordinateurs d'apprendre à partir de données pour effectuer des prédictions ou résoudre des tâches spécifiques, sans programmation explicite [2].

Il existe trois grandes catégories d'apprentissage automatique [2] :

1. Apprentissage supervisé : Après avoir été entraînés sur des données d'entrée associées aux résultats souhaités, les modèles peuvent prédire les résultats pour de nouvelles données inconnues.
2. Apprentissage non supervisé : L'ordinateur ne reçoit que les données brutes et doit identifier lui-même des structures ou des motifs pertinents (souvent par clustering), sans supervision externe. L'apprentissage se fait par observation.
3. Apprentissage par renforcement : C'est une approche où la machine (agent) apprend par interaction avec son environnement, en cherchant à découvrir les actions qui maximisent une récompense.

L'avantage de cette approche est que les valeurs de polarité sont souvent estimées statistiquement, réduisant le travail humain pour la création de ressources et offrant une couverture plus large. Il est important de noter que si les textes utilisés pour l'apprentissage ne sont pas représentatifs des données sur lesquelles le modèle est appliqué, les prédictions peuvent être inexactes [13].

1.8.4 L'approche hybride

L'approche hybride combine des techniques d'apprentissage automatique et des méthodes basées sur le lexique, éventuellement enrichies de règles linguistiques. Elle utilise souvent plusieurs classifieurs de sentiments (basés sur un lexique ou l'apprentissage) en cascade ; si un classifieur ne parvient pas à traiter un texte, le suivant prend le relais, et ainsi de suite jusqu'à classification. Une approche hybride peut exploiter les techniques d'apprentissage automatique pour identifier de nouveaux termes et construire des dictionnaires plus riches et contextuellement spécifiques. Les modèles hybrides donnent souvent d'excellents résultats en termes de précision.

1.9 Méthode d'analyse des sentiments VADER « Valence AwareDictionary and sentiment Reasoner »

Le VADER (Valence Aware Dictionary and sEntiment Reasoner) est un outil d'analyse des sentiments fonctionnant sur des règles, particulièrement conçu pour l'analyse des textes issus des médias sociaux. Il est entièrement open source sous licence MIT. Développé par Hutto et ses collègues en 2014, il prend un texte en entrée et fournit en sortie un dictionnaire de scores pour quatre catégories : Négatif, Neutre, Positif, et Composé (score normalisé) [8].

Cet outil a été largement utilisé dans de nombreux systèmes et présente plusieurs atouts. En effet, les émotions qu'il identifie sont considérées comme de bons indicateurs des besoins : une émotion positive signale qu'un besoin est satisfait, tandis qu'une émotion négative indique qu'il ne l'est pas [20].

VADER calcule un score composé qui indique l'orientation globale du sentiment d'une phrase (positif, négatif ou neutre) en additionnant les scores de valence de chaque mot du lexique, ajustés par des règles spécifiques, puis normalisés entre -1 et +1. Cette méthode utilise des seuils basés sur des proportions de texte pour chaque catégorie :

- Positif $\geq 0,05$
- Neutre entre -0,05 et 0,05
- Négatif $\leq -0,05$.

Cette méthode a été sélectionnée pour être intégrée dans notre application.

1.10 Prédiction de lieux et systèmes de recommandation

De nombreux systèmes de recommandation ont été développés dans le but d'offrir des suggestions personnalisées aux utilisateurs. Ces systèmes visent à extraire (explicitement ou implicitement) les préférences des utilisateurs afin de leur présenter des recommandations plus pertinentes.

L'avènement des technologies de géolocalisation (comme le GPS) a permis aux réseaux sociaux d'intégrer les données de localisation de diverses manières. Les utilisateurs peuvent partager leur position, publier des photos géolocalisées sur les plateformes de médias sociaux, ou commenter un lieu visité.

Étant donné l'importance des commentaires d'utilisateurs sur les lieux qu'ils ont fréquentés, de nombreux systèmes de recommandation s'appuient sur l'analyse de ces commentaires [23].

De plus, la prise en compte du contexte joue un rôle déterminant dans l'amélioration de la qualité des systèmes de recommandation touristiques [1]. Parmi les informations contextuelles, la localisation est l'élément le plus fréquemment utilisé dans les RS touristiques actuels. Des études ont analysé les comportements de personnes visitant des

zones protégées [15] ou utilisé la position actuelle de l'utilisateur pour proposer des recommandations [18] (par exemple, suggestions basées sur les préférences pour les transports en commun et la marche). Le système PSiS [4] recommande des destinations appropriées en fonction du contexte (lieu, heure, vitesse, direction, météo), utilisant l'historique et les préférences touristiques de l'utilisateur pour une meilleure précision.

La majorité des RS touristiques traditionnels s'appuient sur les évaluations numériques des utilisateurs concernant les lieux. Cependant, le texte des commentaires peut contenir plus d'informations riches qu'une simple note, et leur analyse peut significativement améliorer la précision de ces systèmes [23]. Des chercheurs ont analysé les opinions des utilisateurs via des techniques d'exploration de texte [22].

Loh et al. (2003) [14] ont conçu une page de discussion privée pour extraire un vocabulaire spécifique basé sur une ontologie, afin de détecter les préférences de l'utilisateur et faire des recommandations. POST-VIA360 [6], un RS bio-inspiré, vise à fidéliser les touristes après leur première visite en utilisant une ontologie du tourisme et en considérant les visites passées, la localisation actuelle et les aspects sociaux. Plus récemment, Veena (2022) [21] a présenté une méthode de recommandation exploitant les technologies de l'Internet des Objets (IoT).

1.11 Conclusion

En conclusion de ce chapitre, nous avons présenté les principes fondamentaux des systèmes de recommandation, en mettant en lumière leurs principaux types et domaines d'application. Nous nous sommes également intéressés à l'analyse des sentiments, en particulier à la méthode VADER, que nous avons intégrée dans notre application pour enrichir la qualité des recommandations. Enfin, nous avons passé en revue quelques travaux connexes portant sur la prédiction de lieux, afin de situer notre démarche dans le contexte plus large des recherches en systèmes de recommandation.



Chapitre 2:

Conception

Chapitre 02 : Conception

2.1 Introduction

Ce chapitre a pour objectif de décrire les différentes étapes de la conception de notre système de recommandation de lieux utilisant une approche hybride multicritères. Ce système est capable de fournir des recommandations de lieux en fonction des préférences des utilisateurs, en combinant plusieurs méthodes : à base de contenu, filtrage collaboratif, évaluations multicritères, etc. Ce système apprend à classer et de recommander les lieux pertinents en fonction de leurs catégories, de leur proximité géographique, des avis et des commentaires des utilisateurs, etc. Dans ce travail, les données collectées pour tester le système sont analysées à l'aide d'une méthode d'analyse de sentiments (VADER). Dans cette section nous écrivons en détaille la conception du modèle proposé en donnant les détails de chaque module de la conception.

2.2. Architecture générale du système

La Figure 2.1 ci-dessous illustre l'architecture conceptuelle de notre système proposé.

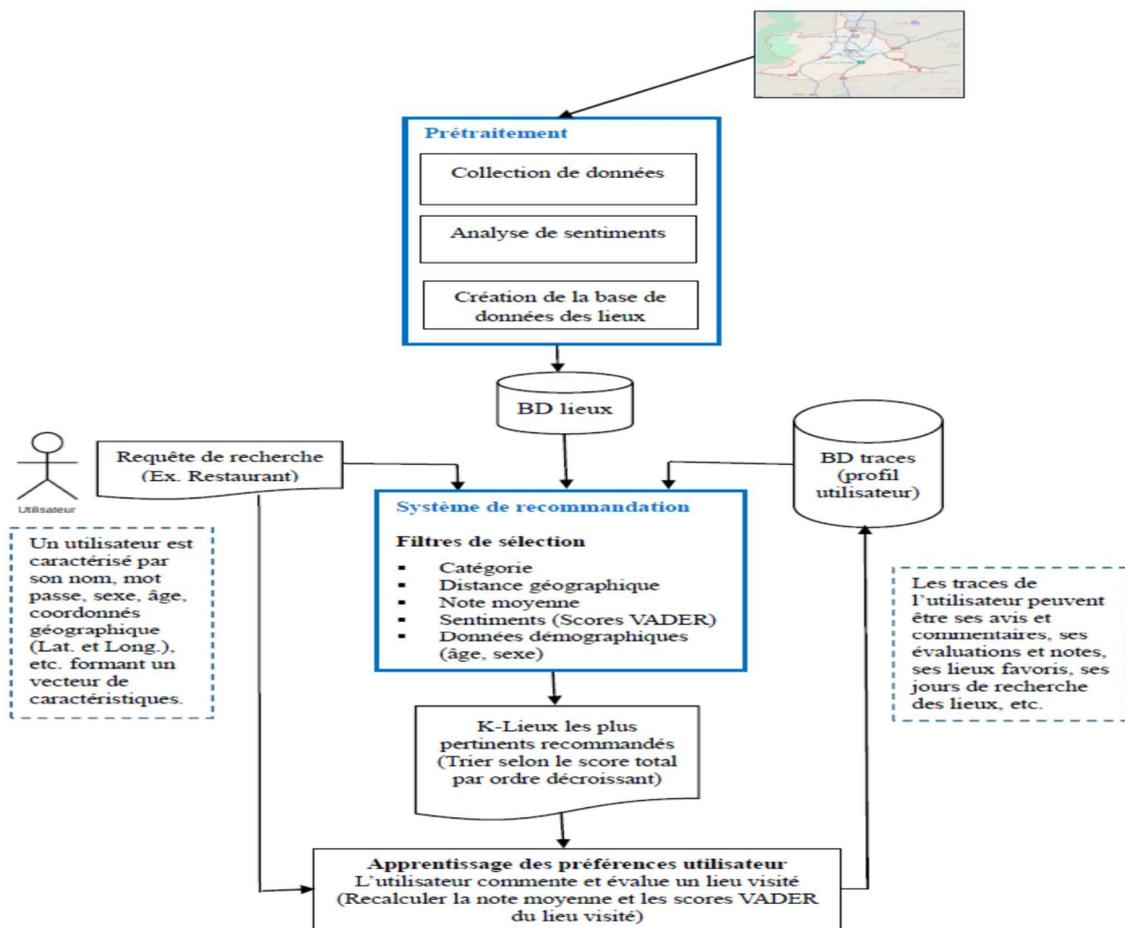


Fig. 2.1. Architecture générale du système proposé

2.2.1 Prétraitement

2.2.1.1 Collection des données

L'ensemble de données que nous avons constitué regroupe divers types de lieux (restaurants, médecins, avocats, etc.) extraits depuis Google Maps. Pour cibler notre étude, nous avons limité la zone géographique à la wilaya de Saïda, afin d'optimiser la pertinence des recommandations proposées. Comme illustré dans la figure suivante, chaque lieu est décrit à travers plusieurs attributs, notamment son nom, sa catégorie, ainsi que sa position géographique (latitude et longitude), entre autres informations utiles à l'analyse. Pour collecter les données relatives aux lieux situés dans la ville de Saïda, nous avons utilisé l'outil Apify, une plateforme d'automatisation web permettant de scraper des données à grande échelle. Grâce à son acteur prédéfini Google Maps Scraper, il a été possible d'extraire des informations pertinentes depuis Google Maps, telles que le nom des établissements, leur adresse, leurs coordonnées géographiques (latitude et longitude), la note moyenne attribuée par les utilisateurs, le nombre d'avis, ainsi que les descriptions et catégories associées. Cette méthode automatisée s'est avérée efficace pour constituer un corpus de lieux variés (restaurants, cafés, hôtels, etc.) à utiliser dans le cadre de notre système de recommandation. Nous avons fait un traitement léger sur notre ensemble de données, en supprimant les lieux en dehors de la wilaya de Saïda. Le corpus de lieux contient environ 2416 lignes.

dPlace	Name	Category	Latitude	Longitude	Rating	Positive Score	Neutral Score	Negative Score
1	Hôtel El Forsane	Hotel	34,8214859	0,1580445	3,6	0,103	0,862	0,035
6	Restaurant - cafétéria	Restaurant	34,851621	-0,062007	5	0,544	0,263	0,193
7	Tipaza Restaurant	Restaurant	34,7614948	0,1458989	4,4	0,341	0,659	0
9	Buffalo	Restaurant	34,8317889	0,1478827	4,5	0	1	0
13	Restaurant marhba	Family restaurant	34,7568375	0,145186	5	0,776	0,224	0
20	مطعم ومقهى السد الاخضر	Restaurant	34,6711878	0,1473559	3	0,712	0,288	0
21	مطعم بوشنة لمشويات المرדومة	Barbecue restaurant	34,5035591	0,8477541	4	0,349	0,651	0
23	Restaurant Miloud	Fast food restaurant	34,8203147	0,1527799	3,8	0,645	0,355	0
24	Chicken House	Pizza restaurant	34,8410314	0,1571644	3,8	0,265	0,735	0
25	Pizzeria	Fast food restaurant	34,8302839	0,1508722	5	0	1	0
26	Restaurant Le Palmier "Ac	Restaurant	34,8281633	0,1508252	4,3	0,308	0,672	0,021
29	خيمة مشوي زقاي لفرع	Barbecue restaurant	34,8235254	0,1547454	4,2	0,284	0,716	0
32	snap food	Pizza restaurant	34,828176	0,1500663	4,3	0,367	0,633	0
36	Pizzeria Eden+	Pizza restaurant	34,7631344	0,1458344	5	0,767	0,233	0
38	Cafeteria Zaoui	Restaurant	34,7607445	0,1483412	5	0,492	0,508	0
39	مطعم خيمة "الميناما"	Family restaurant	34,8055895	0,1449874	4,6	0,231	0,706	0,063
248	Hôtel d'orient	Hotel	34,830767	0,1516206	4,3	0,677	0,323	0
249	Dortoir El-SOROR	Hotel	34,830404	0,1472479	4,4	0,176	0,824	0
250	Dortoir El Zohour	Hotel	34,8317503	0,1443833	3,6	0,792	0,208	0
251	Hotel El Mordjane	Hotel	34,831624	0,1450827	3,2	0,744	0,256	0
252	Hotel Azzouz	Hotel	34,8315533	0,1505712	3,5	0,547	0,453	0
254	Mineral water Hotel	Hotel	34,9328023	0,1804682	3,5	0,249	0,751	0
256	Pharmacie Benmhamed	Pharmacy	34,990663	0,4803853	4,1	0	1	0
257	Pharmacie Rabahia	Pharmacy	34,8779956	0,1530529	4,7	1	0	0
258	Pharmacie Driss Khodja B	Pharmacy	34,8291735	0,1574391	3,9	0,106	0,822	0,072
261	Pharmacie Osman	Pharmacy	34,8296569	0,1534542	4,2	0,25	0,75	0
262	Pharmacie CHIFAA	Pharmacv	34,8372338	0,175068	3	0	0,804	0,196

Fig. 2.2 – Un exemple d'une partie de la collection des données

2.2.1.2 Analyse des sentiments

Dans le cadre de notre système de recommandation de lieux, l'analyse des sentiments joue un rôle fondamental pour enrichir les critères de pertinence au-delà des simples notes numériques. Pour ce faire, nous avons exploité les avis textuels des utilisateurs collectés à partir de Google Maps. Chaque commentaire associé à un lieu a été soumis à une analyse de sentiments à l'aide du lexique VADER (Valence Aware Dictionary and sEntiment Reasoner), un outil efficace et adapté à l'évaluation de textes courts comme les avis en ligne. VADER attribue à chaque commentaire quatre types de scores : positif, négatif, neutre et un score compound, ce dernier représentant une mesure globale de la tonalité du message, variant entre -1 (très négatif) et +1 (très positif).

Pour chaque lieu, l'ensemble des avis disponibles a été analysé, et les scores ont été agrégés afin d'obtenir des valeurs moyennes représentatives. Ces scores sentimentaux ont été intégrés dans notre base de données sous forme de nouveaux attributs, enrichissant ainsi le profil descriptif de chaque lieu. Par exemple, un restaurant dont les commentaires sont majoritairement positifs bénéficiera d'un score compound élevé, ce qui renforcera sa visibilité dans les recommandations. Cette analyse permet de nuancer les évaluations classiques basées uniquement sur la note (rating), en intégrant la dimension qualitative des expériences utilisateurs. Par ailleurs, lors de l'ajout d'un nouvel avis, le système recalcule dynamiquement les scores VADER du lieu concerné, assurant ainsi une mise à jour continue et une fiabilité accrue des données. L'intégration de ces scores dans notre algorithme de recommandation permet donc une approche plus fine et plus personnalisée, en tenant compte à la fois de la popularité objective d'un lieu et de la perception subjective exprimée par les utilisateurs.

Le vecteur de caractéristique de chaque lieu comme il est présenté dans la figure précédente contient le nom, la catégorie, position géographique (latitude et longitude), note et les scores de Vader. Ensuite, pour chaque lieu, nous avons calculé les scores VADER positif, neutre et négatif selon une méthode d'agrégation simple.

- **Score positif**, par exemple, est obtenu en faisant la moyenne des scores positifs de chaque commentaire d'un utilisateur. Formellement, cela peut être exprimé ainsi : $\text{Score positif} = (\sum \text{valeur_positive_commentaire}_i) / N$, où N représente le nombre total de commentaires.

Ci-dessous, un exemple concret tiré de notre base de données. Après avoir appliqué l'analyse VADER aux commentaires associés au lieu «Restaurant-Cafétéria». Le résultat montre que ce lieu présente un score positif moyen de 54.40 %, ce qui reflète une appréciation globalement favorable de la part des utilisateurs.

6	Restaurant - caf��teria	Restaurant	34,851621	-0,062007	5	0,544	0,263	0,193
---	-------------------------	------------	-----------	-----------	---	-------	-------	-------

Fig. 2.3 - Un exemple r  el positif depuis l'ensemble de donn  es.

- **Score n  gatif** est calcul   en proc  dant de la m  me mani  re que pour le score positif. Il s'agit de calculer la moyenne des valeurs n  gatives extraites de chaque commentaire. Formellement, cela s'  crit comme suit :
Score n  gatif = $(\sum \text{valeur_n  gative_commentaire_i}) / N$.

   titre d'illustration, l'analyse des commentaires li  s au restaurant « Pharmacie CHIFFA », r  alis  e avec la m  thode VADER, a r  v  l   un score n  gatif moyen de 19.60 %. Ce pourcentage indique que, bien que le lieu soit globalement bien per  u, une proportion mod  r  e d'avis exprime des aspects n  gatifs ou insatisfaisants.

262	Pharmacie CHIFAA	Pharmacy	34,8372338	0,175068	3	0	0,804	0,196
-----	------------------	----------	------------	----------	---	---	-------	-------

Fig. 2.4 - Un exemple r  el n  gatif depuis l'ensemble de donn  es.

2.2.1.3 Cr  ation de la Base de donn  es

Apr  s l'extraction et le traitement des donn  es depuis Google Maps, celles-ci ont   t   structur  es sous format CSV en vue de leur exploitation. L'ensemble de donn  es a ensuite   t   export   vers une base de donn  es **MySQL**    l'aide de scripts d'insertion automatis  s. Chaque enregistrement a   t   ins  r   dans des tables sp  cifiques, en respectant la structure d  finie au pr  alable (lieux, cat  gories, coordonn  es, avis, etc.). Cette   tape a permis de centraliser les donn  es dans un environnement relationnel, facilitant leur interrogation et leur utilisation dans le syst  me de recommandation. Ainsi, les donn  es sont d  sormais pr  tes pour les traitements analytiques et les recommandations personnalis  es. Le sch  ma ci-dessous de la figure 2.5, repr  sente la conception de notre base de donn  es :

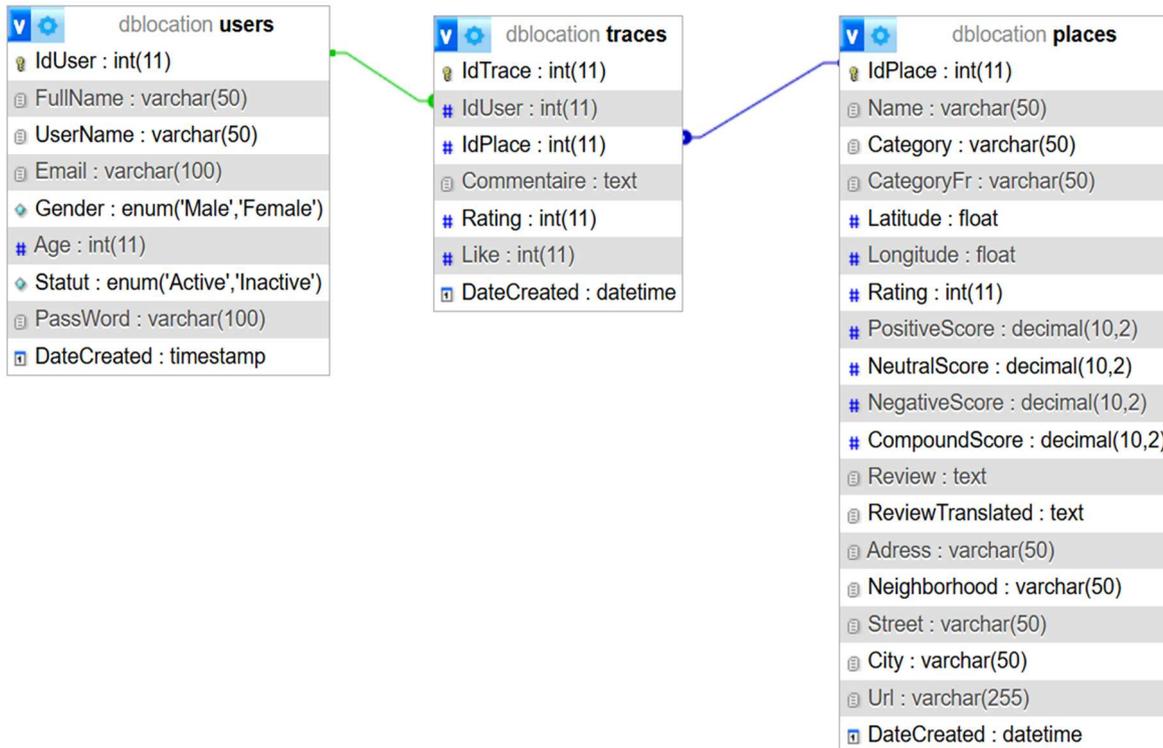


Fig. 2.5 – Création de la base de données

- **La table users** : Cette table contient les informations des utilisateurs de l’application.
- **La table Places** : Cette table stocke les informations relatives aux lieux extraits de Google Maps.
- **La table traces** : Cette table fait le lien entre les utilisateurs et les lieux visités.

2.2.2. Fonctionnement du système

Les informations relatives aux lieux, telles que leur position géographique, leur catégorie, ainsi que les différents scores d’évaluation (notation moyenne, scores VADER issus de l’analyse des commentaires), sont importées dans la base de données de l’application que nous avons développée. Ce système a pour objectif principal de proposer aux utilisateurs des recommandations de lieux pertinents, en se basant sur leur position actuelle, la distance aux lieux environnants, ainsi que la qualité des évaluations attribuées par d’autres utilisateurs. Ces évaluations incluent aussi bien les notes que les commentaires analysés à l’aide de la méthode VADER pour en extraire une polarité (positive, neutre ou négative).

Lorsqu’un utilisateur effectue une recherche, par exemple pour un restaurant, le système sélectionne les établissements répondant à ses critères, puis applique un filtre basé sur la distance, la note globale, la proportion de commentaires positifs, leurs âges et sexes. Une fois le lieu visité, l’utilisateur est invité à évaluer son expérience à travers une note et un commentaire. Ce dernier est ensuite soumis à une analyse sémantique par VADER, permettant de déterminer le score de sentiment correspondant.

En fonction du résultat de cette analyse, le système ajuste dynamiquement ses recommandations. Si le commentaire est jugé positif, le lieu demeure recommandé ; dans le cas contraire, il est exclu des futures suggestions pour cet utilisateur et intégré à une liste noire personnalisée. Ce mécanisme contribue à l'amélioration continue de la pertinence des recommandations.

Par ailleurs, pour enrichir le processus de personnalisation, le système exploite les traces d'interaction des utilisateurs, c'est-à-dire l'historique de leurs recherches, les lieux visités, les commentaires publiés et les notes attribuées. Ces données comportementales permettent au système d'apprendre les préférences individuelles de chaque utilisateur. Ainsi, un lieu régulièrement visité par un utilisateur à des moments précis (par exemple, tous les dimanches) sera proposé de manière anticipée, à condition que ses évaluations restent favorables. Ce fonctionnement adaptatif fait de notre système de recommandation intelligent, capable de s'auto-ajuster en fonction de l'expérience et des retours de ses utilisateurs.

2.2.3 Algorithme de recommandation

Voici un algorithme détaillé pour un système de recommandation de lieux utilisant une approche hybride multicritères, incluant une recherche par catégorie suivie d'un classement basé sur plusieurs critères.

Approche générale : L'algorithme combine trois techniques :

- Filtrage par contenu (catégorie, distance, score VADER, rating)
- Filtrage collaboratif (traces utilisateur)
- Filtrage démographique/contextuel (âge, sexe)

1. Données en entrée

- lat_user, lon_user : position géographique de l'utilisateur
- age_user, gender_user : Profil démographique de l'utilisateur
- category : catégorie recherchée (ex:restaurant,...)
- traces_user : lieux visités, notés, commentés par l'utilisateur
- lieux[] : base de données des lieux avec :
 - lat, lon
 - catégorie
 - rating
 - score VADER (pos, neutre, négatif, compound)

2. Données en sortie

Une liste triée des k lieux les plus pertinents selon les critères de recherches.

3. Étapes de l'algorithme

Étape 1 : Filtrage par catégorie des lieux

On commence par filtrer les lieux pour ne conserver que ceux appartenant à la catégorie recherchée.

```
lieux_filtres = [lieu for lieu in lieux if lieu.categorie == category]
```

Étape 2 : Filtrage par distance minimale

Filtrage par distance minimale entre l'utilisateur et chaque lieu de la base des lieux en utilisant la formule de Haversine :

```
distance = haversine(lat_user, lon_user, lat_lieu, lon_lieu)
distance_score = 1-min(distance/max_distance,1) # Normalisé entre 0 et 1
```

Étape 3 : Note moyenne des utilisateurs

On normalise la note moyenne attribuée par les utilisateurs :

```
rating_score = lieu.rating / 5 # Normalisé entre 0 et 1
```

Étape 4 : Score de sentiment (analyse VADER)

Deux variantes possibles pour le score de sentiment :

Méthode simple (compound) :

```
sentiment_score = (lieu.vader_compound + 1) / 2 # Normalisé entre 0 et 1
```

Méthode pondérée par composantes :

```
sentiment_score = ( lieu.vader_positive * 1.0 + lieu.vader_neutral * 0.5
+ lieu.vader_negative * 0.0)
```

Ce score est déjà compris entre 0 et 1.

Étape 5 : Similarité avec l'historique utilisateur (traces)

On modélise le comportement utilisateur à partir des lieux qu'il a visités :

Créer un vecteur moyen des lieux visités par l'utilisateur (vecteur_user)
éléments : catégorie, score VADER, rating, etc.

Construire un vecteur pour chaque lieu à recommander.

Calculer la similarité cosinus :

```
vecteur_user = moyenne(des caractéristiques des lieux visités)
```

```
similarité = cos_sim(vecteur_user, vecteur_lieu)
```

```
traces_score = normaliser(similarité) # [0, 1]
```

Étape 6 : Score démographique (âge et sexe)

1. Regrouper les utilisateurs similaires (même sexe, classe d'âge)

2. Calculer la moyenne des notes/sentiments pour ce lieu par ces utilisateurs

3. Définir un score démographique :

```
demo_score = moyenne(notes + sentiments par utilisateurs similaires)
```

Étape 7 : Agrégation multicritères - Calcul du score global

Chaque critère est pondéré, et un score final est calculé :

```
final_score=(w1*rating_score +w2 * sentiment_score +w3 * traces_score +w4
* demo_score)
```

Exemple de pondération : W1 = 0.30 (note), w3 = 0.25 (traces), W2 = 0.25 (sentiment), w4 = 0.20 (profil)

Étape 8 : Tri selon final_score et sélection des k meilleurs lieux.

On trie les lieux filtrés selon final_score décroissant

2.3 Conception de notre système

Cette section détaille la conception de notre application en utilisant les diagrammes UML (Unified Modeling Language) pour modéliser les aspects fonctionnels, structurels et dynamiques du système.

2.3.1 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les interactions entre les acteurs externes et le système, illustrant les fonctionnalités offertes par l'application. La figure 2.6 ci-dessous représente le diagramme de cas d'utilisation de notre système.

L'administrateur est le responsable de notre système, et son rôle principal est la mise à jour de la base de données du système. L'ensemble des utilisateurs peuvent visiter notre application, mais une fois leurs inscriptions sont validées ils peuvent consulter les catégories des lieux, rechercher des lieux, commenter et évaluer des lieux visités, voir les détails d'un lieu, etc.

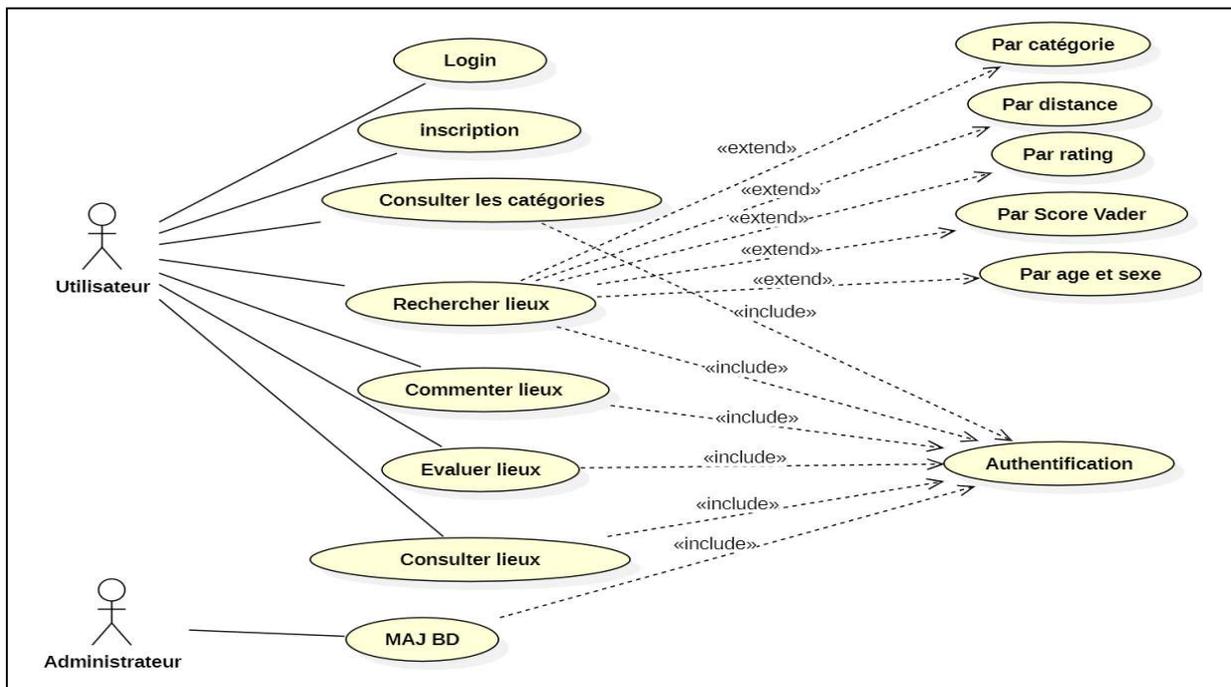


Fig. 2.6 – Diagramme de cas d'utilisation

2.3.2 Diagrammes de classe

Les diagrammes de classe décrivent la structure statique du système, montrant les classes, leurs attributs, leurs opérations et les relations entre elles. La figure 2.7 ci-dessous représente le diagramme de classe de notre système. Le diagramme de classe représente un modèle de données pour une application de recommandation de lieux. Il contient trois classes principales : User, Place et Trace. Les classes User (utilisateur) et Place (lieu) sont liées par la classe d'association Trace, qui modélise l'évaluation d'un lieu par un utilisateur.

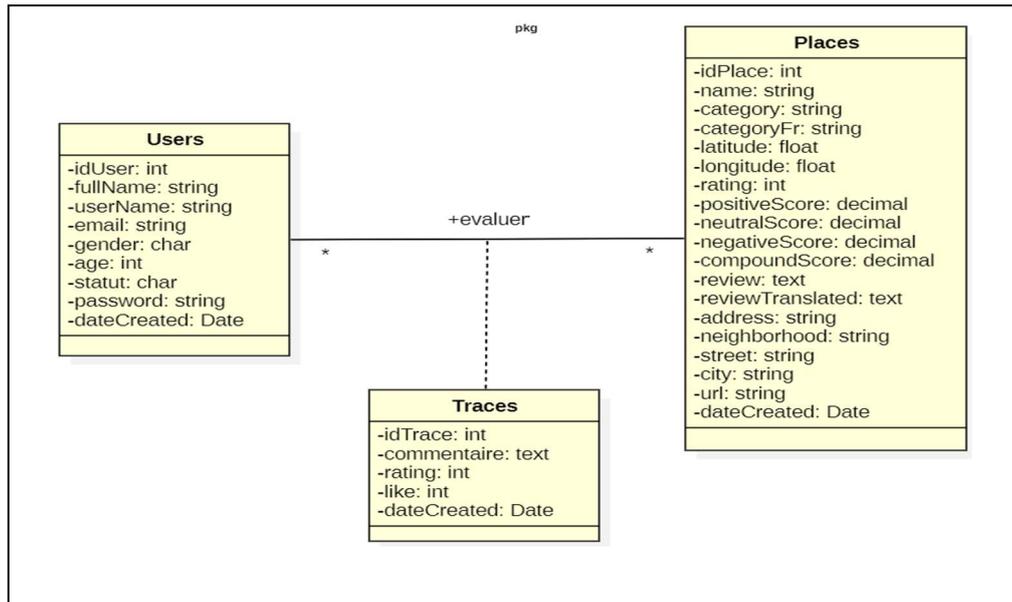


Fig. 2.7 - Diagramme de classe

2.3.3 Diagrammes de séquence

Les diagrammes de séquence décrivent les interactions entre les objets dans le temps pour un cas d'utilisation spécifique. Les figures 2.8, 2.9 et 2.10 ci-dessous représentent trois interactions de l'utilisateur avec notre système. Nous allons illustrer les cas d'utilisations « Login », « Inscription » et « consulter les lieux à proximité ».

Diagramme de séquence « Login »

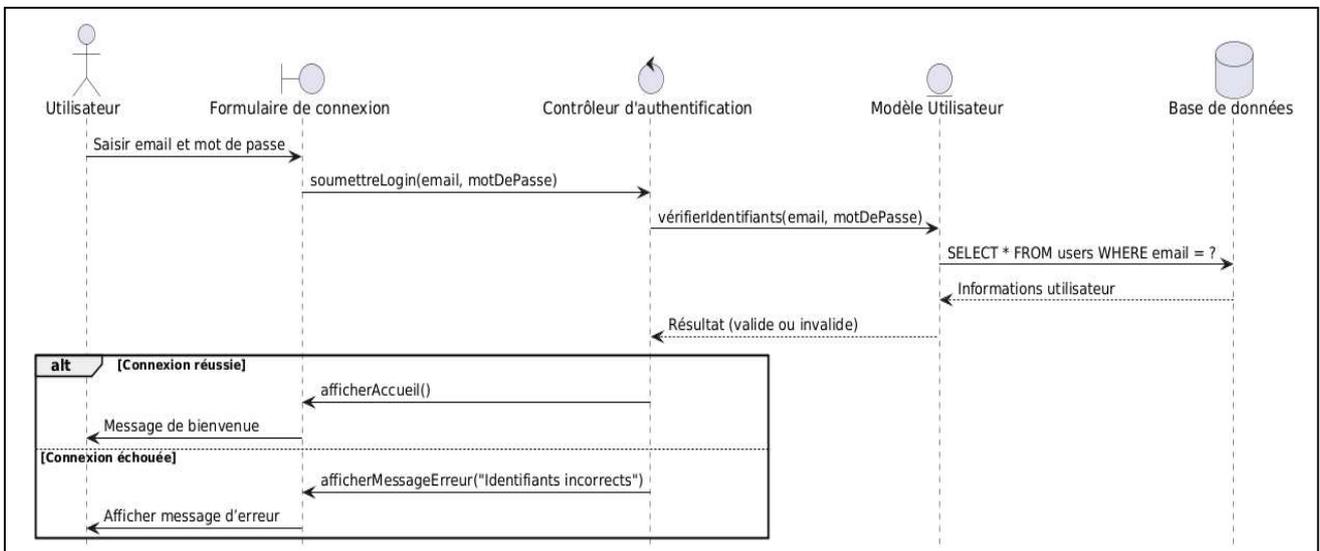


Fig. 2.8 - Diagramme de séquence « Login »

Chaque utilisateur doit remplir le formulaire de connexion avec son nom d'utilisateur et son mot de passe. Le système vérifie l'existence de ce compte dans la base de données, s'il existe alors l'authentification est réussite et affiche un message de bienvenue sinon le système affiche un message d'erreur.

Diagramme de séquence « Inscription »

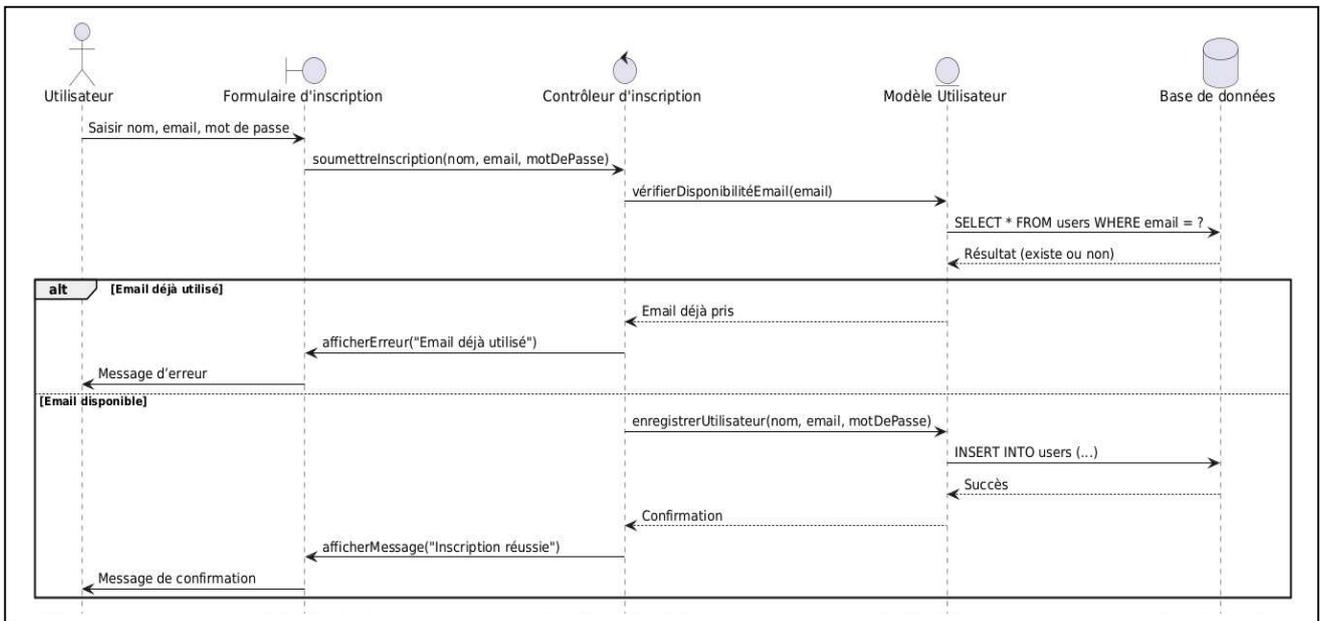


Fig. 2.9 – Diagramme de séquence « Inscription »

Pour s’inscrire, l’utilisateur fournit ses informations (nom, email, mot de passe, âge, sexe, etc.) via un formulaire. Le système vérifie si l’email est déjà utilisé, enregistre les données si tout est valide, et retourne un message de confirmation ou d’erreur.

Diagramme de séquence « consulter les lieux à proximité »

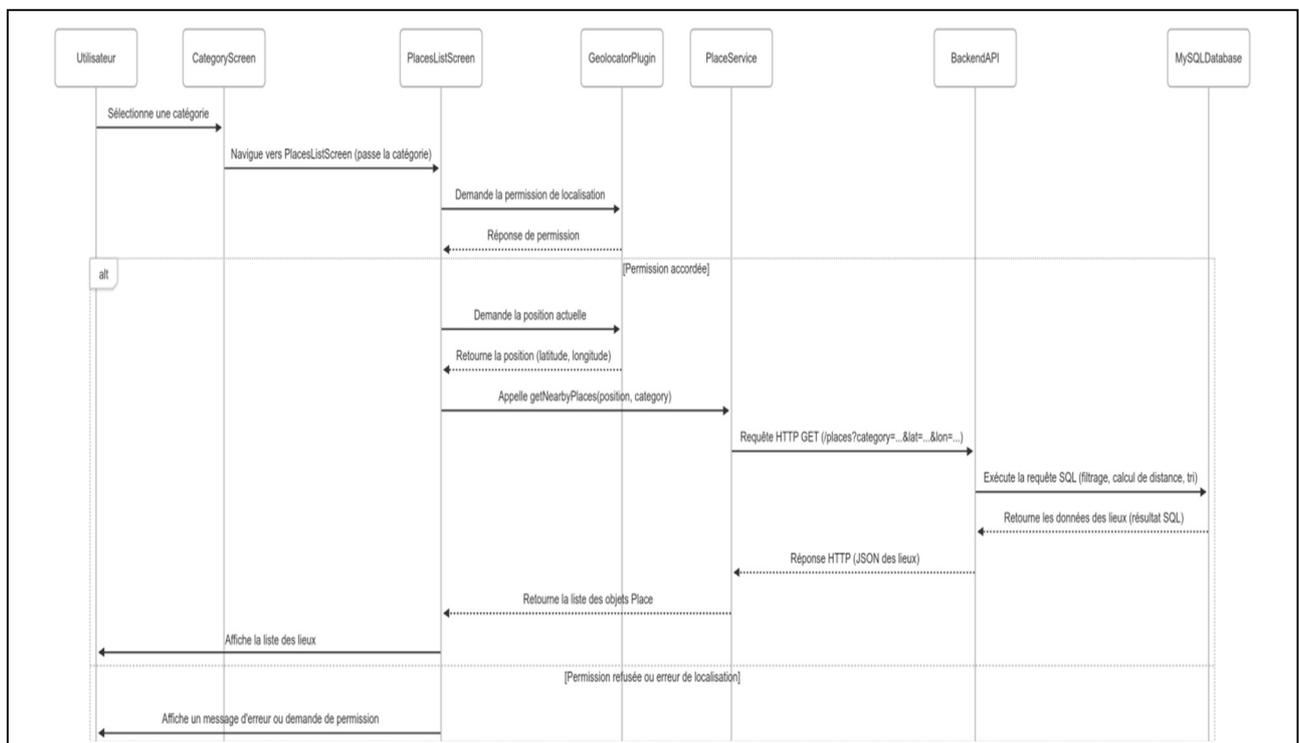


Fig. 2.10 – Diagramme de séquence « consulter les lieux à proximité »

Le diagramme illustre le flux d'interactions lorsque l'utilisateur sélectionne une catégorie. L'écran des catégories navigue vers l'écran de la liste des lieux. Cet écran interagit avec le plugin Geolocator pour obtenir la position. Si la position est obtenue, il appelle le service de lieux, qui communique avec l'API backend. L'API backend interagit avec la base de données MySQL pour récupérer, filtrer et trier les données avant de les renvoyer au service, qui les transmet à l'écran pour affichage à l'utilisateur. Le cas alternatif de refus de permission ou d'erreur de localisation est également représenté.

2.4 Conclusion

Ce chapitre a posé les bases de la conception de notre système de recommandation de lieux, en abordant l'architecture générale et les mécanismes clés permettant de répondre aux besoins des utilisateurs souhaitant découvrir des endroits pertinents dans la région de Saïda, en Algérie. Nous avons d'abord défini le contexte du problème de recommandation de lieux et justifié la nécessité d'une approche adaptée aux spécificités locales et aux données disponibles. Nous avons opté pour un système hybride multicritères qui combine plusieurs dimensions pour affiner les suggestions de lieux. Cette hybridation repose principalement sur :

- La recommandation basée sur la localisation : En utilisant les coordonnées GPS de l'utilisateur, le système identifie et priorise les lieux à proximité géographique.
- La recommandation basée sur le contenu : Le filtrage par catégorie permet à l'utilisateur d'explorer des types de lieux spécifiques (restaurants, hôtels, etc.), ce qui correspond à une forme de recommandation basée sur ses préférences explicites.
- La recommandation basée sur l'utilité/évaluation : L'intégration des scores de sentiment (positif, neutre, négatif, composé) et de l'évaluation générale (Rating) des lieux permet d'introduire une dimension de qualité et de popularité dans le processus de tri et de recommandation.

La combinaison de ces critères permet d'offrir à l'utilisateur une liste de lieux qui ne sont pas seulement proches, mais aussi pertinents par rapport à la catégorie recherchée et jugés favorablement par d'autres (via les avis et évaluations).

En conclusion, ce chapitre a présenté une architecture solide et une approche de recommandation hybride adaptée au contexte de notre projet. Les diagrammes UML ont permis de visualiser cette structure et les interactions clés.



Chapitre 3:

Implémentation et
expérimentation

Chapitre 03 : Implémentation et expérimentation

3.1 Introduction

Dans un paysage numérique en constante évolution, l'accès rapide et intuitif à l'information locale est devenu un facteur clé pour enrichir l'expérience quotidienne des citoyens et des visiteurs. C'est dans ce contexte que s'inscrit le développement de "Saïda Finder", une application mobile novatrice conçue pour transformer la manière dont les utilisateurs explorent et interagissent avec la wilaya de Saïda. Loin d'être une simple énumération de lieux, cette application se positionne comme un guide numérique intelligent, offrant une plateforme centralisée et dynamique pour découvrir une multitude de points d'intérêt, allant des hébergements et options de restauration aux établissements éducatifs, services de santé, lieux de divertissement et administrations publiques, etc. L'ambition première de "Saïda Finder" est de dépasser la fonctionnalité de base de la recherche pour proposer une expérience utilisateur hautement personnalisée. Grâce à l'intégration de fonctionnalités de recherche avancées, à la gestion de profils utilisateurs personnalisables et, de manière significative, à un système sophistiqué d'analyse de sentiment des avis, l'application vise à fournir des recommandations non seulement pertinentes mais également contextualisées. Elle permet aux utilisateurs de bénéficier des retours d'expérience d'autres membres de la communauté et de partager leurs propres perceptions, créant ainsi un écosystème d'information riche et collaboratif.

Ce chapitre est dédié à une exploration approfondie du processus d'implémentation et des résultats de l'expérimentation de notre application « Saïda Finder ». Nous détaillerons l'environnement de développement choisi, les langages de programmation et les technologies qui ont été les piliers de sa construction. Une attention particulière sera portée à la description des interfaces graphiques, illustrant le parcours utilisateur et l'esthétique visuelle qui caractérisent l'application.

3.2 Environnements de développement et langages utilisés

Le succès de notre application mobile repose sur une sélection rigoureuse et stratégique de technologies et d'environnements de développement. Le choix de chaque outil a été guidé par la nécessité de construire une application à la fois robuste, performante, sécurisée et offrant une expérience utilisateur fluide et agréable. L'environnement de développement regroupe l'ensemble des outils de gestion informatiques qui participent à la conception et la création de notre application. De nos jours beaucoup d'outils d'implémentation des sites web sont mis à la disposition des utilisateurs, on va donc citer ceux qu'on a utilisés. On cite également les langages de programmation utilisés afin de réaliser ce projet.

- **PHP (Hypertext Preprocessor)**

PHP a été choisi comme langage de script côté serveur pour la construction de l'API RESTful de "Saïda Finder". Sa maturité, sa vaste communauté, et sa compatibilité avec MySQL en font un choix solide pour le développement de services web robustes. Le PHP a été choisi comme étant un langage de programmation libre, spécialement adopté pour

créer des pages web actives via un serveur HTTP et ainsi agir comme n'importe quel langage décodé de façon locale. Aussi, PHP permet le codage absolu orienté objet. [38].

- **MySQL**

MySQL, un système de gestion de base de données relationnelle open-source, a été la colonne vertébrale pour le stockage structuré de toutes les données persistantes de l'application. Sa fiabilité, ses performances avérées pour la gestion de grands volumes de données et ses capacités de jointure ont été des atouts majeurs. [32]

- **Sublime Text Editor**

Sublime text se définit comme un « éditeur de texte sophistiqué pour le code, le balisage et la prose » initialement, en janvier 2008, on est maintenant à sa quatrième version majeure. Il prend en charge Mac, Windows et Linux. Sublime Text ressemble essentiellement à un couteau suisse qui peut être appliqué à n'importe quel cas d'utilisation ou difficulté impliquant la manipulation de texte. Contrairement aux autres éditeurs de texte, Sublime Text permet d'exécuter le code directement dans l'application. Ceci spécialement efficace pour les projets simples comme les petits scripts Bash ou Python [33].

- **HTML (Hypertext Markup Language) & CSS (Cascade Style Sheet)**

HTML et CSS décrivent comment une page web est présentée. HTML est l'ossature du web. Il arrange les différents éléments d'une page web ainsi le CSS permet de donner un style visuel pour ces éléments [34].

- **Leaflet**

Leaflet est une bibliothèque fondamentale JavaScript à code source ouvert pour les cartes interactives adaptées aux téléphones portables. Leaflet ne dépasse pas les 42 Ko de JS, elle offre toutes les fonctionnalités auxquelles ont recours certains développeurs [35]. Bien qu'elle n'ait pas été directement intégrée à l'interface Flutter finale (où Google Maps Flutter serait plus approprié pour une carte native), sa connaissance représente une expertise précieuse dans le domaine de la cartographie web. Elle pourrait être envisagée pour une interface web associée, par exemple, pour visualiser les lieux et leurs données sur une carte administrative en ligne.

- **Apify**

Apify est une plateforme de scraping web et d'automatisation qui permet de collecter des données à grande échelle. Son utilisation aurait été cruciale pour l'étape initiale de constitution de la base de données des lieux d'intérêt de Saïda. En automatisant la collecte d'informations (noms, adresses, catégories, etc.) à partir de sources publiques en ligne, Apify aurait considérablement accéléré le remplissage initial de la base de données MySQL, garantissant ainsi une richesse de données dès le démarrage de l'application [36].

- **JavaScript & jQuery**

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, généralement connu comme le langage de script des pages, utilisé pour rendre les pages interactives. JS permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des

animations 2D/3D, des menus vidéos défilants, etc. C'est la troisième couche des technologies standards du web, les deux premières sont HTML et CSS. Pour rendre nos pages interactives nous avons utilisé le langage JavaScript et sa bibliothèque JQuery [37].

- **Bootstrap**

Bootstrap est une infrastructure de développement frontale, gratuite et open source pour l'innovation des sites et d'applications Web. Bootstrap repose sur HTML, CSS et JavaScript (JS) pour rendre le développement des sites et d'applications réactives et tout-mobile plus facile [38].

- **GOOGLE COLAB, Python, NumPy, Pandas, VADER**

Ces technologies ont constitué le cœur de l'implémentation de l'analyse de sentiment, une fonctionnalité clé pour raffiner les recommandations de lieux. Le processus a impliqué plusieurs étapes :

- **Python:** Python est le langage de programmation le plus connu entre les informaticiens ces dernières années. Il est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. Python permet aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font [40]. Langage de programmation principal pour le traitement des données textuelles et l'intégration des modèles d'analyse de sentiment. Sa richesse en bibliothèques a facilité l'ensemble du pipeline.
- **NumPy:** Bibliothèque fondamentale pour le calcul scientifique en Python, utilisée pour des opérations numériques efficaces sur les données d'avis, notamment pour la manipulation de tableaux de données et l'optimisation des calculs [41].
- **Pandas :** Essentielle pour la manipulation et l'analyse de données structurées. Pandas a permis le prétraitement des avis utilisateurs : chargement, nettoyage (suppression des caractères spéciaux, ponctuation), normalisation (mise en minuscule), et structuration des textes pour les rendre exploitables par l'algorithme de sentiment.
- **VADER (Valence AwareDictionary for Sentiment Reasoning):** Un outil spécifiquement conçu pour l'analyse de sentiment basée sur des lexiques (dictionnaires de mots avec des scores de polarité associés). VADER est particulièrement adapté aux textes courts et informels, comme les commentaires de médias sociaux, ce qui le rend pertinent pour les avis utilisateurs. Il fournit des scores de polarité (positif, négatif, neutre) ainsi qu'un score composé. VADER a été intégré au backend PHP (via un appel à un service Python ou une implémentation équivalente) pour évaluer la polarité des commentaires soumis par les utilisateurs, permettant ainsi de classer les avis et, à terme, d'influencer le score global d'un lieu [42].
- **GOOGLE COLAB:** Un environnement de notebook Jupyter basé sur le cloud. Il permet aux utilisateurs d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement adapté à l'apprentissage automatique (machine learning), à l'analyse de données (data analysis) et à l'éducation (e-learning). Google Colab a été un outil précieux pour le prototypage rapide, le test et l'expérimentation de l'algorithme VADER. Il a permis d'itérer rapidement sur le prétraitement des données, de visualiser les résultats de l'analyse de sentiment sur

des échantillons d'avis, et d'optimiser l'intégration avant le déploiement final du module dans le backend PHP [39].

• Flutter

Flutter, le framework de développement d'interface utilisateur de Google, a été le choix principal pour la création de l'application mobile. Ses avantages sont multiples :

- Développement Cross-Platform: Permet de construire des applications natives et performantes pour Android (et potentiellement iOS) à partir d'une seule base de code Dart, réduisant considérablement le temps et les coûts de développement.
- Hot Reload et Hot Restart: Accélèrent le cycle de développement en permettant aux développeurs de voir instantanément les changements apportés au code.
- UI Riche et Expressive: Les widgets de Flutter, hautement personnalisables, ont permis de concevoir des interfaces utilisateur visuellement attrayantes, avec des animations fluides et des designs complexes.
- Performances NATIVES: Flutter compile le code en code machine natif, garantissant des performances élevées et une expérience utilisateur réactive.

Flutter a été utilisé pour la conception de tous les écrans (Login, Register, Profile, Categories, Places List, User Comments Log), la gestion de la navigation entre ces écrans, l'intégration des APIs backend (PHP) pour la récupération des données et la soumission des actions utilisateurs, ainsi que l'affichage des données en temps réel provenant de Firebase.

• Firebase Core, Firebase Authentication, Cloud Firestore

Faisant partie de l'écosystème Google Firebase, ces services ont complété l'architecture de notre application, offrant des solutions évolutives et gérées pour des fonctionnalités critiques :

- FirebaseCore: Le module de base indispensable pour l'intégration de tout service Firebase dans l'application Flutter. Il initialise la connexion avec le projet Firebase.
- FirebaseAuthentication (FirebaseAuth): Gère l'authentification des utilisateurs (inscription, connexion, réinitialisation de mot de passe) de manière sécurisée et sans effort. FirebaseAuth prend en charge diverses méthodes d'authentification (e-mail/mot de passe, Google, Facebook, etc.), réduisant ainsi la complexité de la gestion des sessions utilisateurs pour les développeurs et offrant une expérience de connexion fiable aux utilisateurs.
- Cloud Firestore: Une base de données NoSQL flexible et évolutive, optimisée pour la synchronisation des données en temps réel. Cloud Firestore a été utilisé spécifiquement pour le stockage et la synchronisation des commentaires des utilisateurs. Sa capacité à mettre à jour instantanément les données sur tous les clients connectés a permis d'implémenter un "Journal des Commentaires" où les avis sont visibles par tous les utilisateurs dès leur soumission, favorisant ainsi une interaction dynamique et en temps réel.

- **AppsGeyser**

AppsGeyser est une plateforme de création d'applications sans code, permettant de convertir des sites web en applications Android ou de créer des applications simples à partir de modèles. Son inclusion dans la liste des outils pourrait indiquer une phase initiale de prototypage ou la création d'une version simplifiée de l'application pour des tests rapides sur des appareils Android. Avant d'investir pleinement dans le développement plus complexe et personnalisé avec Flutter, AppsGeyser aurait pu servir à valider l'idée de base ou à obtenir des retours préliminaires sans écrire de code, confirmant ainsi le besoin d'une application dédiée [43].

3.3 Les Interfaces Graphiques du Système

L'application mobile "Saïda Finder" a été conçue avec une attention particulière à l'expérience utilisateur et à l'esthétique visuelle. Chaque écran est le fruit d'un design réfléchi, visant à offrir une navigation intuitive et un accès aisé aux informations.

3.3.1 Écran de Connexion (Login Screen)

- ✓ **Objectif** : Permettre aux utilisateurs déjà enregistrés d'accéder de manière sécurisée à l'application. Permet aux visiteurs de notre application de s'authentifier via un nom d'utilisateur et un mot de passe avec une option d'inscription s'ils ne possèdent pas de compte.
- ✓ **Éléments Clés** :
 - Deux champs de saisie distincts pour l'adresse e-mail et le mot de passe, avec une validation en temps réel pour guider l'utilisateur.
 - Un bouton d'action clair et bien visible pour la connexion.
 - Un lien "Mot de passe oublié ?" (implicite) et un lien explicite invitant les nouveaux utilisateurs à s'inscrire, dirigeant vers l'écran d'inscription.



Fig. 3.1 - Écran de Connexion (Login Screen)

3.3.2 Écran d'Inscription (Register Screen)

- ✓ **Objectif** : Offrir aux nouveaux utilisateurs une procédure simple et rapide pour créer un compte. Les internautes peuvent créer des comptes, en remplissant les champs du formulaire avec les informations qui les correspondent (nom d'utilisateur, âge, email, sexe et mot de passe).
- ✓ **Éléments Clés** :
 - Champs de saisie pour le nom d'utilisateur, l'adresse e-mail et le mot de passe (potentiellement avec confirmation du mot de passe).
 - Un bouton d'inscription proéminent.
 - Un lien "Déjà un compte ? Connectez-vous" pour revenir à l'écran de connexion.



Fig. 3.2 – Écran d'Inscription (Register Screen)

3.3.3 Écran du Profil Utilisateur (User Profile Screen)

- ✓ **Objectif** : Afficher les informations personnelles de l'utilisateur et offrir des options de personnalisation de profil.
- ✓ **Éléments Clés** :
 - Affichage clair du nom d'utilisateur et de l'ID utilisateur, essentiel pour les applications collaboratives où l'identification est importante.
 - Circle Avatar Interactif pour l'Image de Profil: C'est une fonctionnalité clé. L'utilisateur peut cliquer sur cet avatar pour :
 - Choisir une nouvelle image depuis la galerie de son appareil.
 - Prendre une photo directement avec l'appareil photo.
 - Supprimer l'image de profil existante.
 - Boutons d'action pour la déconnexion et d'autres paramètres éventuels.



Fig. 3.3 – Écran du Profil Utilisateur (User Profile Screen)

3.3.4 Écran des Catégories (Category Screen)

Cet écran est le tableau de bord principal de l'application après l'authentification et est le point de départ pour l'exploration des lieux. Cet écran est constitué de :

- ✓ **Introduction et Logo** : se situe en haut de l'écran, cette section dynamique affiche le logo de l'application (par exemple, un logo stylisé de Saïda ou un symbole lié à la découverte) et un message de bienvenue alternant fluidement entre l'arabe et le français.
- ✓ **Barre de Recherche** : se situe juste sous l'introduction, une barre de recherche moderne et stylisée est implémentée. Elle permet aux utilisateurs de filtrer les catégories de lieux en temps réel. Au fur et à mesure que l'utilisateur tape, la liste des catégories se met à jour instantanément, affichant uniquement celles qui correspondent à la saisie. Cette fonctionnalité améliore considérablement l'accessibilité et la rapidité de navigation.
- ✓ **Liste des Catégories** : Le corps principal de l'écran présente une liste déroulante verticale des catégories de lieux disponibles (ex: "Hôtels", "Restaurants", "Pharmacies", "Administration", etc.). Chaque catégorie est représentée par une Card ou un Tile stylisé, incluant une icône représentative pour une reconnaissance visuelle rapide et une meilleure navigation. Un clic sur une catégorie dirige l'utilisateur vers l'écran de liste des lieux correspondant.

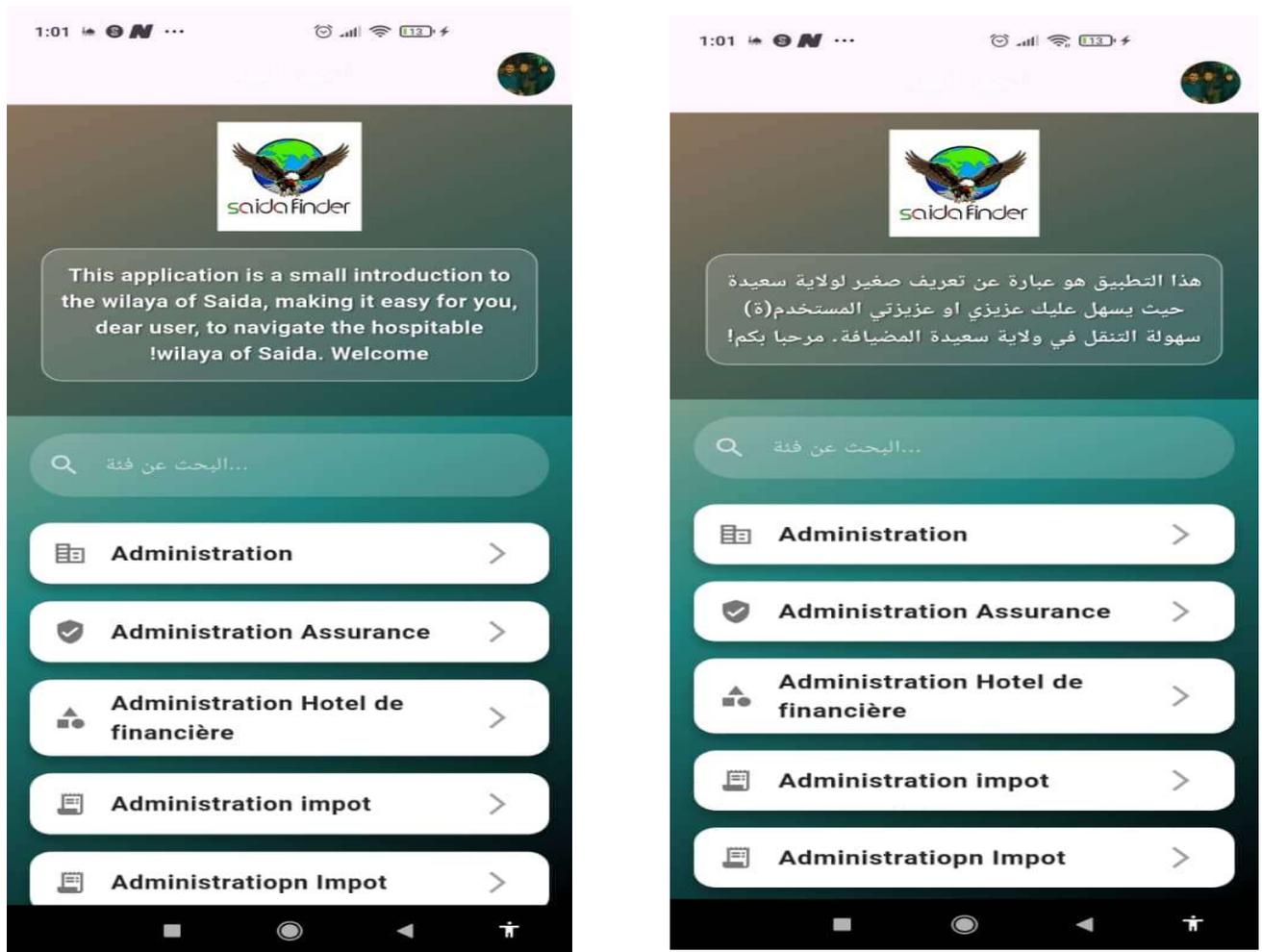


Fig. 3.4 – Écran des Catégories (Category Screen)

3.3.5 Écran de Liste des Lieux (Places List Screen)

- ✓ **Objectif** : Afficher dynamiquement une liste de lieux recommandés, filtrés en fonction de la catégorie sélectionnée par l'utilisateur et de sa position géographique actuelle, etc. Cet écran présente des lieux jugés pertinents selon plusieurs critères, notamment la catégorie choisie, la proximité, la note moyenne, ainsi que les scores d'analyse de sentiment issus des avis, etc.
- ✓ **Éléments Clés** :
 - Une liste déroulante des lieux appartenant à la catégorie choisie (par exemple, toutes les administrations de Saïda).
 - Chaque élément de la liste affiche des informations pertinentes pour chaque lieu : son nom, son adresse, sa distance, et potentiellement une évaluation moyenne.

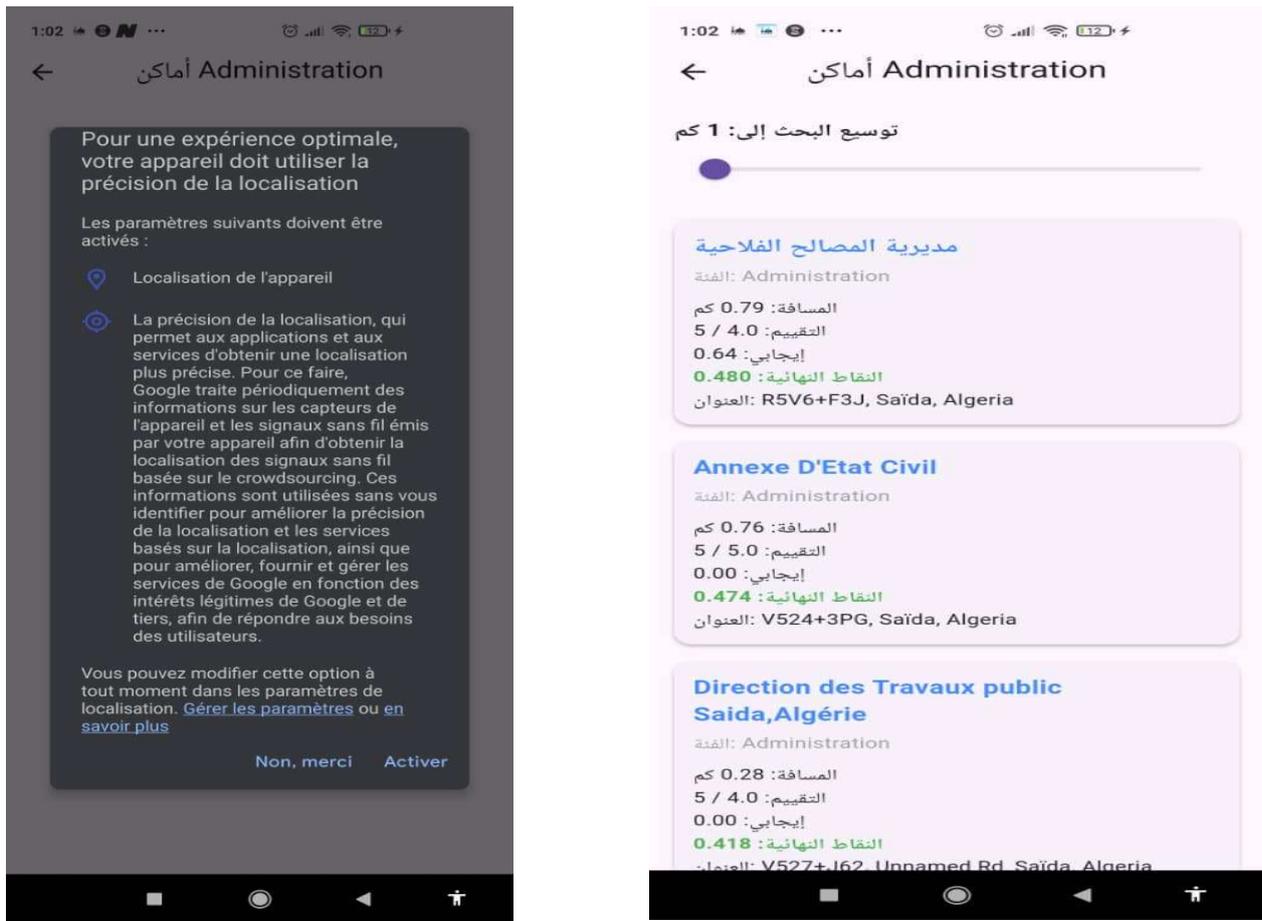


Fig. 3.5 – Écran de Liste des Lieux (Places List Screen)

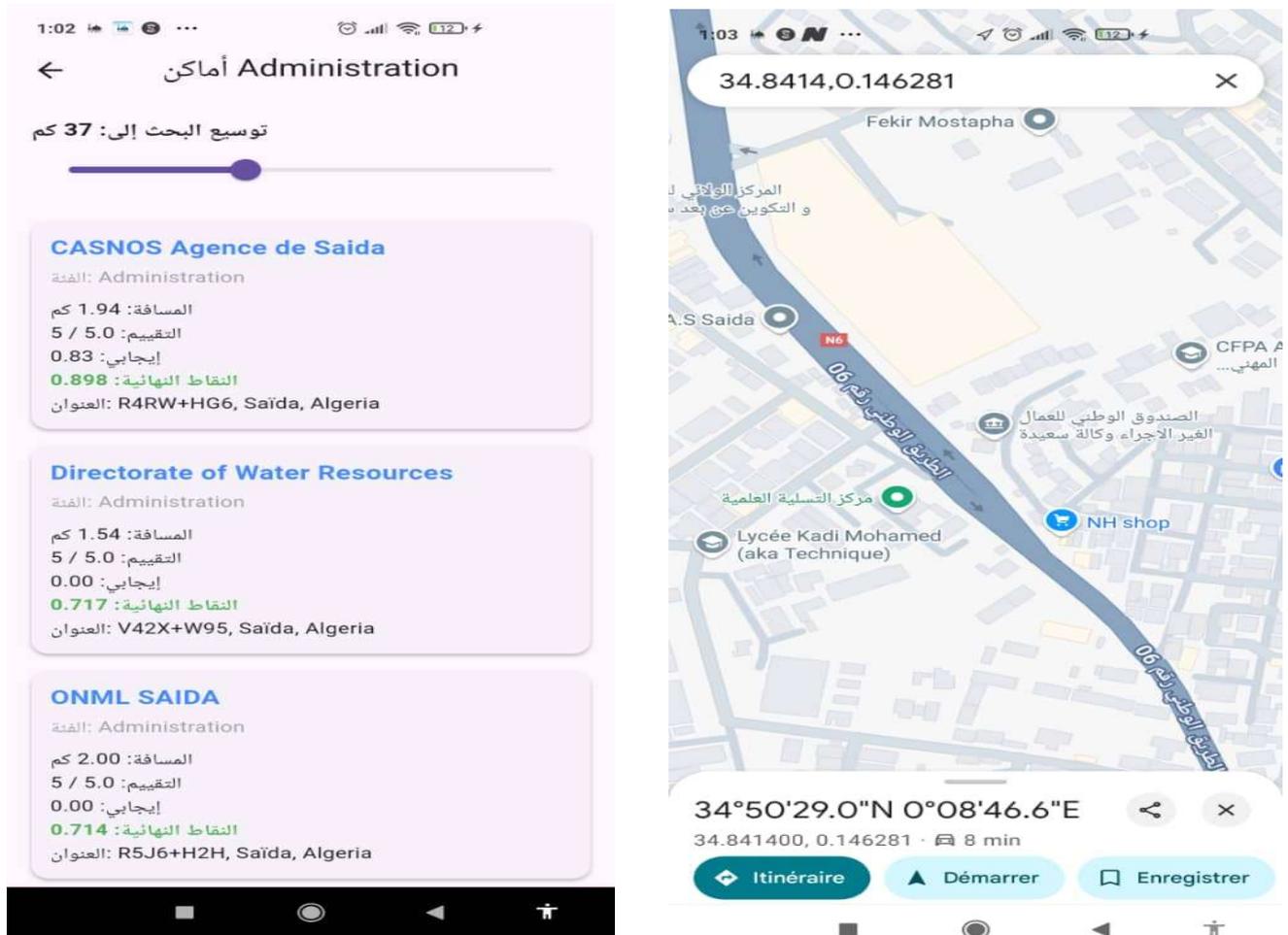


Fig. 3.6 – Écran de Liste des Lieux et carte google map

3.3.6 Écran du Journal des Commentaires Utilisateurs (User Comments Log Screen)

- **Objectif** : Offrir une vue d'ensemble de tous les avis soumis par les utilisateurs de l'application, favorisant la transparence et la participation communautaire. Après avoir visité cet endroit, l'utilisateur peut évaluer et commenter cet endroit. Ce journal constitue une source essentielle pour le calcul des scores de sentiment.
- **Éléments Clés** : Une liste déroulante de cartes individuels, chaque carte représentant un commentaire distinct. Chaque carte de commentaire affiche clairement :
 - Le nom du lieu auquel le commentaire se réfère.
 - Le nom d'utilisateur (ou l'ID utilisateur) de la personne qui a soumis l'avis.
 - Les notes dévaluation
 - Le texte intégral du commentaire.
 - Un lien pour accéder à la carte google map du lieu choisi.

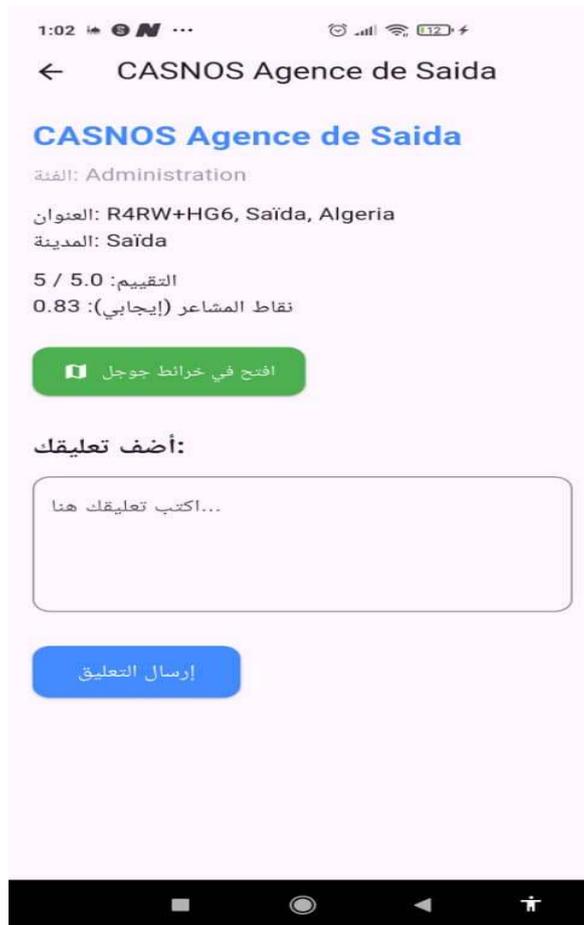


Fig. 3.7 – Écran du Journal des Commentaires Utilisateurs (User Comments Log Screen)

En conclusion, le développement de l'application « Saïda Finder » a abouti à une solution mobile complète et fonctionnelle, répondant à l'objectif de faciliter la découverte et l'exploration de la wilaya de Saïda. Les résultats obtenus et les leçons apprises tout au long du processus sont encourageants et mettent en lumière la pertinence de l'architecture et des technologies choisies.

3.4 Expérimentation et discussion

Pour évaluer rigoureusement la qualité des recommandations générées par notre système nous avons utilisés la mesure de précision qui s'agit de l'une des métriques les plus couramment utilisées utilisés dans le domaine de l'évaluation des systèmes de recommandation. La précision P se calcule selon la formule suivante : $P = \frac{nrp}{ntr}$, où nrp = nombre de recommandations pertinentes et ntr = nombre total des recommandations.

Dans le cadre de cette évaluation, une expérimentation en conditions réelles a été menée avec la participation d'une dizaine d'utilisateurs. Ces derniers ont été invités à créer un compte sur notre application et à l'utiliser en conditions réelles dans la ville de Saïda. Chaque participant a testé le système dans divers emplacements de la ville, recevant ainsi différentes suggestions de lieux. Pour chaque utilisateur, quatre (04) lieux ont été

recommandés. Ces derniers ont ensuite attribué une note à chaque lieu visité, reflétant leur niveau de satisfaction. Ces retours ont permis d'identifier les recommandations jugées pertinentes (note ≥ 3). Les résultats obtenus sont présentés dans le tableau ci-dessous :

Id-Utilisateur	1	2	3	4	5	6	7	8	9	10
Pertinents dans top-4	2	4	3	2	3	4	3	4	4	3
Précision	0,50	1,00	0,75	0,50	0,75	1,00	0,75	1,00	1,00	0,75

Tableau 3.1 : Résultats de l'expérimentation

Enfin, ces résultats ont été agrégés pour analyser la performance générale du système de recommandation en termes de pertinence. La précision moyenne obtenue pour les lieux recommandés par le système s'élève à 80%, ce qui constitue un résultat très encourageant. Ce taux indique que la majorité des recommandations proposées ont été jugées pertinentes par les utilisateurs, ce qui témoigne de la capacité du système à répondre efficacement à leurs préférences et attentes.



Conclusion Générale

Conclusion Générale

Ce document a exploré le parcours de développement et d'implémentation d'un système de recommandation de lieux hybride multicritères, spécifiquement conçu pour la wilaya de Saïda. Dans un contexte où la dépendance aux technologies numériques ne cesse de croître, les systèmes de recommandation et les applications mobiles basées sur la géolocalisation sont devenus des outils indispensables pour guider les utilisateurs dans un environnement riche en informations et en choix.

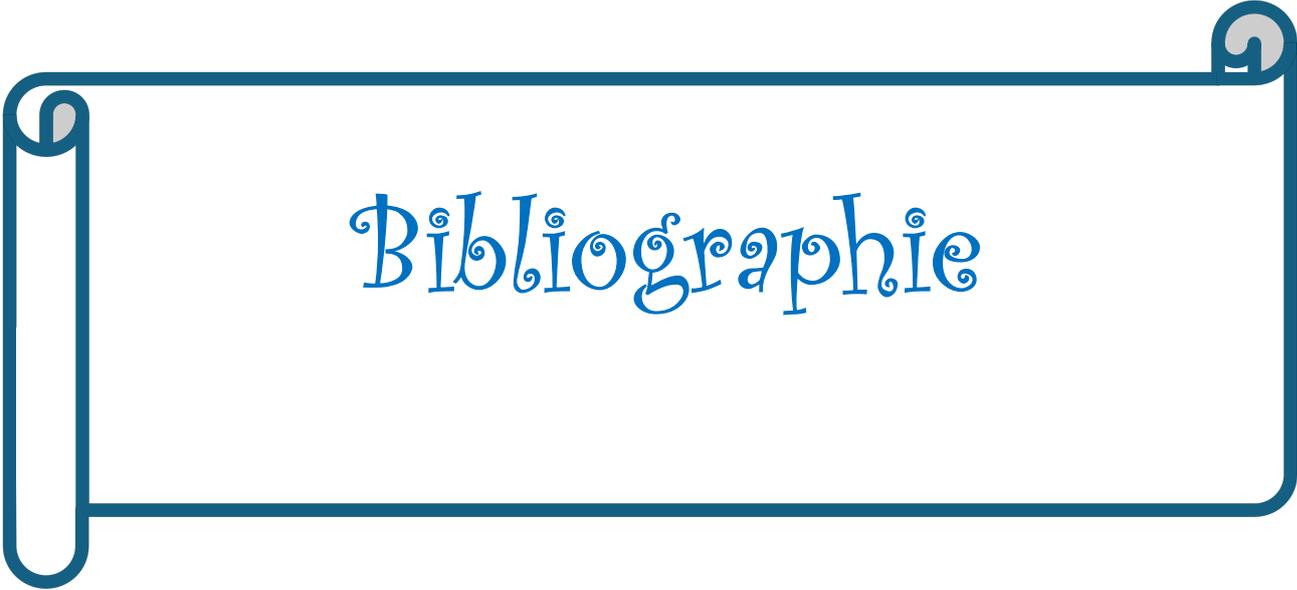
L'objectif principal de ce projet était de construire une solution innovante permettant aux résidents et aux visiteurs de Saïda de découvrir des lieux pertinents de manière efficace et personnalisée. Pour y parvenir, l'étude a adopté une approche combinant plusieurs techniques de recommandation et intégrant l'analyse de sentiment des commentaires utilisateurs, dans le but d'améliorer la précision et la pertinence des suggestions.

Nous avons passé en revue les fondements théoriques des systèmes de recommandation et leurs différentes typologies, ainsi que les méthodologies d'analyse de sentiment. Nous avons ensuite détaillé la conception de notre système, en mettant en lumière l'architecture choisie, les algorithmes employés et les options technologiques qui en ont constitué la colonne vertébrale.

Enfin, une part significative de ce rapport a été consacrée à l'application mobile "Saïda Finder", qui incarne la concrétisation des concepts théoriques. Nous avons présenté l'environnement de développement, les langages et les technologies utilisés. Les interfaces graphiques ont été décrites avec soin, conçues pour offrir une expérience utilisateur intuitive et esthétique.

Les résultats obtenus ont démontré que "Saïda Finder" est une application mobile bien conçue, offrant une expérience utilisateur riche et fonctionnelle pour la découverte et la navigation dans la wilaya de Saïda.

Malgré les succès enregistrés, des voies d'amélioration futures restent ouvertes. Il est envisagé d'explorer des modèles d'apprentissage automatique plus avancés pour l'analyse de sentiment, d'étendre le support multilingue.



Bibliographie

Bibliographie

- [1] Zahra ABBASI-MOUD, Hamed VAHDAT-NEJAD et Javad SADRI. « Tourism recommendation system based on semantic clustering and sentiment analysis ». In : *Expert Systems with Applications* 167 (2021), p. 114324.
- [2] Larbes ABDELKRIM, Amrani OKBA et Khantoul BILEL. « Une Approche deep learning pour l'analyse des sentiments ». In : (2021).
- [3] Mehdi Hosseinzadeh AGHDAM. « Context-aware recommender systems using hierarchical hidden Markov model ». In : *Physica A : Statistical Mechanics and its Applications* 518 (2019), p. 89-98.
- [4] Ricardo ANACLETO et al. « Mobile application to provide personalized sightseeing tours ». In : *Journal of Network and Computer Applications* 41 (2014), p. 56-64.
- [5] Taushif ANWAR et V UMA. « Comparative study of recommender system approaches and movie recommendation using collaborative filtering ». In : *International Journal of System Assurance Engineering and Management* 12.3 (2021), p. 426-436.
- [6] Ricardo COLOMO-PALACIOS et al. « Towards a social and context-aware mobile recommendation system for tourism ». In : *Pervasive and Mobile Computing* 38 (2017), p. 505-515.
- [7] Gina GEORGE et AnishaM LAL. « Review of ontology-based recommender systems in e-learning ». In : *Computers & Education* 142 (2019), p. 103642.
- [8] Clayton HUTTO et Eric GILBERT. « Vader : A parsimonious rule-based model for sentiment analysis of social media text ». In : *Proceedings of the international AAAI conference on web and social media*. T. 8. 1. 2014, p. 216-225.
- [9] INFOpedia - VADER Sentiment Analysis. 2014. URL : <https://cmci.colorado.edu/infopedia/pmwiki.php?n=Analytics.VADERSentimentAnalysis>.
- [10] Marilyn LATOUR. « Analyse de sentiments dans les textes économiques : un exemple d'application chez ReportLinker ». In : () .
- [11] Stanley LOH et al. « A tourism recommender system based on collaboration and text analysis ». In : *Information Technology & Tourism* 6.3 (2003), p. 157-165.
- [12] Tinghuai MA et al. « Social network and tag sources based augmenting collaborative recommender system ». In : *IEICE transactions on Information and Systems* 98.4 (2015), p. 902-910.
- [13] VidishaM PRADHAN, Jay VALA et Prem BALANI. « A survey on sentiment analysis algorithms for opinion mining ». In : *International Journal of Computer Applications* 133.9 (2016), p. 7-11.
- [14] Pierre RICCI, Sibylle BUI et Claire LAMINE. *Repenser la protection des cultures : innovations et transitions*. Editions Quae, 2011.

- [15] Claire B SMALLWOOD, Lynnath E BECKLEY et Susan AMOORE. « An analysis of visitor movement patterns using travel networks in a large marine park, northwestern Australia ». In : *Tourism Management* 33.3 (2012), p. 517-528.
- [16] John K TARUS, Zhendong NIU et Abdallah YOUSIF. « A hybrid knowledgebased recommender system for e-learning based on ontology and sequential pattern mining ». In : *Future Generation Computer Systems* 72 (2017), p. 37-48.
- [17] Adithya THADURI, Uday KUMAR et Ajit Kumar VERMA. « Computational intelligence framework for context-aware decision making ». In : *International Journal of System Assurance Engineering and Management* 8.4 (2017), p. 2146-2157.
- [18] Gytis TUMAS et Francesco RICCI. « Personalized mobile city transport advisory system ». In : *Information and communication technologies in tourism 2009* (2009), p. 173-183.
- [20] Karsten TYMANN et al. « GerVADER-A German Adaptation of the VADER Sentiment Analysis Tool for Social Media Texts. » In : *LWDA*. 2019, p. 178-189.
- [21] Veena Prasad VEMURI et al. « Analyzing the use of internet of things (IoT) in artificial intelligence and its impact on business environment ». In : *Materials Today : Proceedings* 51 (2022), p. 2194-2197.
- [22] Xiaolin ZHENG et al. « Topic tensor factorization for recommender system ». In : *Information Sciences* 372 (2016), p. 276-293.
- [23] Xiaoyao ZHENG et al. « A tourism destination recommender system using users' sentiment and temporal dynamics ». In : *Journal of Intelligent Information Systems* 51.3 (2018), p. 557-578.
- [24] Adomavicius, G., & Tuzhilin, A. (2005). toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* , 17, no 6, 734-749.
- [25] Ait Ahmed, N., & Driss Khodja, A. (2018). *Système de Recommandation de Cours à Base d'Ontologie*.
- [26] Bastin, J. (2020). *Etude des systèmes de recommandations et mise en pratique des algorithmes*.
- [27] Goldberg, D., Nichols, D., OKI, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM* , 35, no 12, 61-70.
- [28] Rao, k. (2008). Application domain and functional classification of recommender systems--a survey. *DESIDOC Journal of Library & Information Technology* , 28, no 3, 17.
- [29] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* , 2009, no1, 421425.
- [30] Nguyen, A., Denos, N., & Berrut, C. (2005). *Cartes de communautés pour l'adaptation interactive de profils dans un système de filtrage d'information*. INFORSID.

- [31] php. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/>
- [32] MySQL. URL : <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-mysql/>.
- [33] sublime. URL : <https://kinsta.com/fr/blog/comment-utiliser-sublimetext/>.
- [34] html css. URL : <https://www.futura-sciences.com/tech/definitions/internet-css-4050/>.
- [35] leaflet. URL : <https://leafletjs.com/>.
- [36] apify. URL : <https://www.capterra.fr/software/150854/apify>.
- [37] javascript. URL : https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
- [38] bootstrap. URL : <https://www.lemagit.fr/definition/Bootstrap>.
- [39] google colab. URL : <https://research.google.com/colaboratory/faq.html>.
- [40] python. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-celangage-informatique/>.
- [41] numpy. URL : <https://numpy.org/doc/stable/user/whatisnumpy.html>.
- [42] vader. URL : <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>.
- [43] VidishaM PRADHAN, Jay VALA et Prem BALANI. «A survey on sentiment analysis algorithms for opinion mining ». In : International Journal of Computer Applications 133.9 (2016), p. 7-11.